Automatic Classification of Speech & Music in Digitized Audio

by

Muhammad Kashif Saeed Khan

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

In Partial Fulfillment of the Requirements
for the Degree

MASTER OF SCIENCE

IN

Information & Computer Science

KING FAHD UNIVERSITY
OF PETROLEUM AND MINERALS

Dhahran, Saudi Arabia

May 2005

Dedicated to

My Beloved Parents

and

My

Maternal Uncle

ACKNOWLEDGEMENTS

In the name of Allah, the Most Gracious and the Most Merciful

All praise and glory goes to Almighty Allah (Subhanahu Wa Ta'ala) who gave me the courage and patience to carry out this work. Peace and blessings of Allah be upon His last Prophet Muhammad (Sallulaho-Alaihe-Wassalam) and all his Sahaba (Razi-Allaho-Anhum) who devoted their lives towards the prosperity and spread of Islam.

First and foremost gratitude is due to the esteemed university, the **King Fahd University of Petroleum and Minerals** for my admittance, and to its learned faculty members for imparting quality learning and knowledge with their valuable support and able guidance that has led my way through this point of undertaking my research work.

My deep appreciation and heartfelt gratitude goes to my thesis advisor **Dr. Wasfi Ghassan Al-Khatib** for his constant endeavour, guidance and the numerous moments of attention he devoted throughout the course of this research work. His valuable suggestions made this work interesting and knowledgeable for me. Working

with him in a friendly and motivating environment was really a joyful and learning experience.

I extend my deepest gratitude to my thesis committee members Dr. Muhammad Sarfraz and Dr. Moustafa Elshafei for their constructive and positive criticism, extraordinary attention and thought-provoking contribution in my research. It was surely an honor and an exceptional learning to work with them.

Acknowledgement is due to my senior fellows Mohammad Moinuddin (Moin BHAI) and Saad Azher for helping me on issues relating to LATEX, MATLAB, and Neural Networks. I will not forget the support, extraordinary attention, and guidance of Moin BHAI in my research.

Sincere friendship is the spice of life. I owe thanks to my house mates, colleagues and my friends for their help, motivation and pivotal support. A few of them are Khawar Khan, Moin BHAI, Saad Azhar, Zeeshan Muzaffar, Aiman Rasheed, Syed Adnan Yusuf, Syed Adnan Shahab, Imran Naseem, Mudassir Masood and many others; all of whom I will not be able to name here. They made my work and stay at KFUPM very pleasant and joyful.

Family support plays a vital role in the success of an individual. I would like to thank my parents, siblings, my maternal uncle and other family members including all my uncles ,aunts and my loving cousins; from the core of my heart. Their prayers and encouragement always help me take the right steps in life.

May Allah help us in following Islam according to Quran and Sunna! *(*Aameen)

Contents

List of Tables

List of Figures

THESIS ABSTRACT

Name: Muhammad Kashif Saeed Khan

Title: Automatic Classification of Speech & Music in Digitized Audio

Degree: MASTER OF SCIENCE

Major Field: Information & Computer Science

Date of Degree: May 2005

The importance of automatic classification between speech signals and music signals has evolved as a research topic over recent years. The need to classify audio into categories such as speech or music is an important aspect of many multimedia document retrieval systems. Several approaches have been previously used to discriminate between speech and music data. In this thesis, we propose the use of the mean and variance of the discrete wavelet transform, variance of mel-frequency cepstral coefficients, RMS of lowpass signal, and difference of maximum and minimum of zero-crossings in addition to other features that have been used previously for audio classification. We have used Multi-Layer Perceptron (MLP) Neural Networks, Radial basis Functions (RBF) Neural Networks, and Hidden Markov Model (HMM) as classifiers. We have also proposed an algorithm to improve the classification accuracy when MLP is applied on audio samples of longer durations. Our experiments have shown encouraging results that indicate the viability of our approach.

Keywords: *Speech music classification, audio signal processing, audio features, neural networks, HMM, fuzzy c-mean.*

King Fahd University of Petroleum and Minerals, Dhahran.
May 2005

ازداد الإهتمام في الآونة الآخيرة بمسألة القدرة على تمييز و تصنيف الإشارات الصوتية إلى إشارات موسيقية وإشارات خطابية. لقد أصبح هذا التصنيف جزءا مهما من انظمة قواعد بيانات مستندات متعددة الوسائط. و قد تمّ القيام بهذا التصنيف في السابق بطرق متعدّدة. ندرس في هذه الرسالة استخدام المعدل mean ودرجة الإنحراف القياسي standatd deviation للتحويل المويجي المتقطع Discrete Wavelet Transform ودرجة الإنحراف القياسي لعوامل ميل التردّدية Mel-Frequency Cepstral Coefficients وجذر المعدل التربيعي للاشارة المخفضة Root Mean Square of a Lowpass Signal والفرق بين الحد الأقصى والحد الأدنى لعبور الصفر zero crossings إضافة لخصائص أخرى استخدمت سابقا. و قد تمّ دراسة طرق التصنيف التالية: الشبكة العصبية التصورية متعددة الطبقات Multi-Layer Perceptron Neural Network والشبكة العصبية ذات الدوال القاعدية الإشعاعية Radial Basis Functions Neural Network و نماذج ماركوف المخفية Hidden Markov Models . كما اقترحنا خوارزمية جديدة لزيادة دقّة تصنيف الشبكة العصبية التصورية متعددة الطبقات للإشارات الصوتية الطويلة. نتائج التجارب التطبيقية أثبتت نجاعة طرقنا المقترحة.

Chapter 1

Sounds and Computing

1.1 Introduction

"Strange, dear..... but true, dear..."

Spinning the radio dial, you hear little snatches of sound:

"...from the BBC world news..."

In a few seconds, or even fractions of a second, you can tell from the sound whether it is a news anchor person, a talk show, or music. What is really intimidating is that, in the space of those few seconds, you effortlessly recognize enough about the vocal personalities and musical styles to tell whether or not you want to listen! Not only can you hear whether those snippets of sound contain music, you can even guess the instruments, hear the notes and lyrics, and recognize melodies in the presence of a

great deal of competing noise.

The problem is that, in a sense, one man's data may well be another man's music. How can one reliably determine whether music is present without a multitude of other sensations that more or less describe all the things that music is? It would certainly seem to help if one could first recognize the constitutive components - the timbre of an instrument, the rhythmic tapping of a drummer, popular melodies, and so on, to know whether or not the sounds make music. One could reasonably conclude that a "proper" music classifier ought to weigh a collection of sensibilities to come up with a believable response.

Similarly, there are many things we listen to and listen for in speech - not simply the textual content of the message, of course, but also the ways in which it was said: for example, who spoke (age, gender, identity), or what expression, intonation, or emotional affect was used. We can tell in an instant whether the voice is that of a happy, angry, or sad person, whether it belongs to a little girl or a big man, to someone who is sick, tired, drunk, or by hearing the trace of an accent, whether the speaker grew up in Karachi or Lahore. Qualities like these are all natural indices to envision using to browse or manipulate large amounts of speech.

In the natural world, sound is the predominant communication modality. It is

the primary medium for the human communication, and even parrots can whistle and mimic speech with only a minimal vocal tract, a brain the size of a pea, and no lips. All this is vaguely disturbing with respect to today's computers: they are adapting exuberantly to graphics, text and spreadsheets, but are unfit for the real acoustic world.

The question is: How can we build machines that make wiser use of sound?
The answer, of course, is that machines need to understand sound, and to operate all along the continuum between a sound wave and representations of its content. Human hearing shows a remarkable ability to "diagnose" sound. In most respects we have barely begun to understand the working of the human auditory system well enough to emulate them in machines. Surely we want interfaces that can listen, understand, and converse naturally with us, and models of sound that are relevant to our sensibilities. While much is known about the nature of many kinds of sounds and about the techniques for processing them, little of that knowledge has found its way into day-to-day systems.

Most importantly, the ability to operate intelligently on sound ultimately depends on sensing structure in the wave that describes meaningful aspects of the source. Our inability to build machines that can do this is partly why sound has been so poorly used in interfaces [1].

1.2 Problem Statement

The exponential growth of the Internet and the latest advances in networking and compression technologies have made huge amounts of audio data easily available. It is not unlikely that in the near future, on-line music services will overtake the usual distribution of audio stored on physical media. Currently, browsing and management of audio data rely mostly on textual information attached manually, which is an extremely time consuming task. Furthermore, this information is often incomplete or not available at all.

Audio signal classification is a field of research that has historically been explored in specific areas like speech recognition, with less work done on the general problem. Other problems in the field have been researched as well, but the main direction has been toward speech applications. The general problem of classification of audio signals into different categories has now reached new levels of interest. Today, classification of audio signals into speech, music, noise, silence, etc. is used as a preprocessing step in almost every audio signal processing application. But the problem that researchers are facing is to come up with audio features that can easily classify audio signals into different categories with higher classification accuracy.

1.2.1 Applications of Audio Signal Classification

Audio signal classification applications are potentially far reaching and relevant. With the fast growth of multimedia repositories, in general, and audio data in specific, the development of technologies for spoken document indexing and retrieval is in full expansion. Audio data sources range from broadcast radio and television, to the humongous volumes of recorded material in different forms, such as tapes and digital audio stored on the Web. Speech/Music classification is an important task in multimedia indexing. It is usually the first step before any further processing on audio data. Some of the applications of the speech/music classification includes:

- *Music reduction/removal from useful documentaries:* One may consider this function as a first step towards the Islamization of media containing useful material, yet is loaded with foreground and/or background music segments. Proper classification of speech & music signals makes it feasible for the media editor to remove all segments identified as "Music" from the documentaries, which, as a result, contributes positively in the full Islamization of such material.

- *Automatic Speech Recognition:* Broadcast radio feed may contain music segments in between different programs. Identifying the "Speech" segments will give more reliable data to the Automatic Speech Recognizer (ASR), which contributes to the minimization of word error rates, out of vocabulary words,

and eliminates unnecessary computations on non-speech data.

- *Context-based Indexing and Retrieval:* After Speech/Music classification process, once can give meaningful descriptions such as speech, music, silence, etc to different segments of the audio data. Such indexing will support querying speech segments only, or music segments only for that matter, from a multimedia database perspective.

- *Speaker recognition:* Extracting speech from the audio signals may enable speaker recognition techniques for identifying and tracking specific speakers for indexing or security purpose.

- *Improving audio coding:* Classifying audio data into speech, music, and silence can be useful in the process of decreasing the bit rate for silence segments and hence improve audio coding.

- *Improving compression techniques:* Some signal compression techniques are more suitable for speech signals, whereas other compression techniques may be more appropriate for music. By automatically determining the audio signal, the appropriate compression technique can be applied.

- *Hearing Instrument:* Automatically adapting a hearing instrument for various listening situations (silence, speech, noise, music, wind, etc.) would free users from manually having to change the mode of the instrument using a push

button located on the hearing instrument, a task that is sometimes problematic for many hearing instrument users.

1.3 Organization of the Thesis

Each subsequent chapter studies a particular aspect of the problem of automatic classification of speech and music. This thesis is organized as follows: we first survey up to date research work that has been carried out in music/speech classification. Then we present an overview of some commonly used audio features and the audio features that we have used in this work. After that we provide a detailed overview of how features are extracted in this work and a comparison of features highlighting the contribution of each feature in the process of classification is presented. In Chapter 5 three different classification frameworks are used to carry out the classification process, and results for each framework are presented and compared. We follow this by introducing our developed software that is used for audio classification in Chapter 6. Finally, a summary highlighting the interesting aspects of our work is presented and possible directions for future work are mentioned.

Chapter 2

Literature Review

In many applications there is a strong interest in classifying audio signals. A variety of systems for audio classification have been proposed and implemented in the past for the needs of various applications. We present some of them in the following paragraphs, permitting a methodological comparison with the techniques proposed in this thesis. We also report their performance for related comparisons. However, due to the lack of a standard test data set, it is not possible to confidently point to this or that work being superior to the others.

Saunders [2] proposed a technique for discrimination of audio as speech or music using the energy contour and the zero-crossing rate. This technique was applied to broadcast radio divided into segments of 2.4 sec which were classified using features extracted from intervals of 16 ms. Four measures of the skewness of the distribution

of the zero-crossing rate were used with a 90% correct classification rate by using multivariate Gaussian classifier. When a probability measure on signal energy was added a performance of 98% is reported. Saad et al [3] have presented a technique to automatically classify audio signal into either speech or music. From 200 frames each of length around 20 ms with adjacent frames overlap one-half of a frame (i.e. 10 ms) they extracted five features[1]:

1. Percentage of Low Energy Frames,
2. Roll Off Point of the Spectrum,
3. Spectral Flux,
4. Zero-Crossing Rate,
5. Spectral Centroid.

Saad et al have proposed an algorithm in which speech/music classification is performed by using average percentage deviation which is calculated by percentage deviation of each feature relative to the maximum deviation of that feature. If it is less than a particular threshold value it is labeled as speech otherwise it is labeled as music. They have reported 94.25% accuracy.

[1]Some of these features will be explained in the next chapter.

Scheirer and Slaney [4] used thirteen features, of which eight are extracted from the power spectrum density, for classifying audio segments into speech and music.

1. 4 Hz modulation energy,
2. Percentage of Low Energy Frames,
3. The Roll Off Point of the Spectrum & it's Variance,
4. The Spectral Centroid & it's Variance,
5. The Spectral Flux & it's Variance,
6. The Zero-Crossing Rate & it's Variance,
7. The Cepstral Residual & it's Variance,
8. Pulse metric.

Of the thirteen, five are "variance" features, consisting of the variance in a one-second window of an underlying measure which is calculated on a single frame. They have log transformed all the thirteen features to improve their spread and conformity to normal distributions. As a classification framework they have investigated four different classifiers:

1. Multi-dimensional Gaussian maximum a posteriori (MAP) estimator,
2. Gaussian mixture model (GMM) classification,

3. Spatial partitioning scheme based on k-d trees,

4. Nearest-neighbor classifier in depth.

A correct classification percentage of 94.2% is reported for 20 ms segments and 98.6% for 2.4 sec segments.

Carey et al [5] presents a comparison of several of the different features, some of them already used in [2, 4], and tested the same data by using Gaussian Mixture Models (GMM) as a classifier and Expectation Maximization (EM) algorithm for training. The following features were used for classification:

1. Cepstral & Delta Cepstral Coefficients,

2. Amplitude & Delta Amplitude,

3. Pitch & Delta Pitch,

4. Zero-Crossing Rate & Delta Zero-Crossing Rate.

Separate experiments were carried out in combination of a feature and its derivative. The best performance resulted from using the cepstra and delta cepstra which gave an equal error rate (EER) of 1.2%. Carey et al extended their work in [6] and again used Cepstral coefficients, amplitude, and pitch features along with GMM. In [6] they proposed two approaches to combine the features to improve overall performance. The first approach uses separate GMM classifiers for each feature type

and fuses the outputs of the classifiers. The second approach combines different features into a single vector prior to modeling the data with a GMM. According to the authors, significant improvements in performance have been observed and an equal error rate of 0.7% has been achieved.

To efficiently index the soundtrack of multimedia documents, it is necessary to extract elementary and homogeneous acoustic segments. Pinquier et al [7, 8, 9] have explored such prior partitioning which consists of detecting audio signals as speech/non-speech and music/non-music by using GMM. For speech detection, cepstral coefficients, entropy modulation and 4 Hz modulation energy were used. For music detection, spectral coefficients, number of segments and segment durations were used. They have reported 93.9% accuracy for speech and 89.9% accuracy for music. Chou and Gu [10] have proposed an approach for robust singing signal detection applied to applications of audio indexing in multimedia databases. The following set of features were used along with GMM for classification:

1. 4 Hz modulation energy
2. Harmonic coefficients
3. 4 Hz harmonic coefficients
4. Mel-Frequency Cepstral Coefficient (MFCC)

5. Log energy

A two stage speech/music discrimination method has been proposed. In the first stage, discrimination is carried out for singing and non-singing signals and in the second stage discrimination between speech and music is done over pre-filtered non-singing segments.

Harb and Chen [11] used first order sound spectrum's statistics as feature vectors. Spectral components of the audio signal were extracted using the Fast Fourier Transform (FFT) with a Hamming window of 30 ms width and a 20 ms overlap. The spectrum is further filtered conforming to the Mel Scale to obtain a vector of 20 Spectral coefficients every 10 ms. Gaussian Mixture Models (GMM) were used for the classification of speech and music but instead of using Expectation Maximization (EM) algorithm or the Gaussian probability density function to estimate the mixture parameters and to calculate the likelihoods, the authors have used a Neural Network (NN) to estimate the probability of each mean/variance model. The NN used is a Multi Layer Perceptron (MLP) with the error back propagation training algorithm and the sigmoid function as an activation one. They have achieved 96% classification accuracy for context-dependent problems and 93% for context-independent ones. In [12], Harb and Chen have investigated audio for indexing purposes and proposed an algorithm that needs no training phase, as in Gaussian Mixture Models based algorithms. It classifies audio signals into 4 classes: speech, music, silence,

and other. Different features were used for different classes like for Silence: Energy level and ZCR, and for Speech/Music: Silence Crossing Rate (SCR) and Frequency Tracking (FT). Classification is achieved by thresholding these features. They have reported 90% classification accuracy.

The possibility to discriminate between speech and music signals by using features based on low frequency modulation has been investigated by Karnebck [13]. Three different low frequency modulation parameters, 4 Hz amplitude and standard deviation, 4 Hz normalized amplitude, and 2-4 Hz normalized amplitude have been extracted and tested by using GMMs. Classification accuracy of 93.6% have been reported. Karnebck extended his research in [14] by comparing the Low Frequency Modulation Amplitude & Deviation (LFMAD) feature, with MFCC and used the GMM as a classification framework. According to Karnebck LFMAD performs better than MFCC but the best results were obtained when these two features were combined in a mixed feature.

Wang et al [15] present a simple approach to classify speech and music in which the proposed modified low energy ratio is first extracted as the only feature and then the system applies the Bayesian MAP (Maximum a posteriori Probability) classifier to decide the audio class. Around 97% of classification accuracy has been reported. El-Maleh et al [16] have focused on frame level narrow band speech/music

discrimination by using four feature sets for experimentation:

1. Line Spectral Frequencies (LSF)
2. Differential LSF
3. LSF with Higher Order Crossings
4. LSF with Linear prediction zero crossing ratio

He used two different classification algorithms: a quadratic Gaussian classifier and a k-nearest neighbor classifier. The k-nearest neighbor classifier gave the best results of 80.85% accuracy. Panagiotakis and Tziritas [17] have developed a system which first segments audio signals and then classifies them into one of the three main categories: speech, music, and silence. They have proposed an algorithm for classification based on RMS and Zero-Crossings. A measure of signal amplitude for a given segment is used for testing the signal presence. This is an estimate of signal amplitude as a weighted sum of mean and median of the RMS. Once it is decided that a signal is present, the speech/music discrimination takes place. They have reported around 95% of classification accuracy.

Beierholm and Baggenstoss [18] have demonstrated the application of the class-specific features approach for the problem of discriminating between speech and music. They have used different features for each class of audio, like for speech:

Linear Predictive Coefficients (LPC), Autocorrelation Function (ACF) values, and Log-Area Ratio (LAR) coefficients. For music, they have used the Power Spectrum and Residual Energy. They have used Probability Density Function (PDF) projection theorem along with Class-specific feature method for classification. The class-specific density functions were estimated using Gaussian mixture HMMs. They have reported 100% classification accuracy. In [19] Goodwin and Laroche have described a two-stage audio segmentation algorithm in which signal features are first extracted and transformed via Linear Discriminant Analysis (LDA) to optimize cluster scatter and then clustered using Dynamic Programming (DP). They have demonstrated the application of the LDA-DP segmentation algorithm to speech/music discrimination.

In [20], Balabko has developed an algorithm for speech and music classification based on the analysis of the speech and music modulation spectrum. According to Balabko, speech modulation spectrum has a typical wide peak at frequencies from 2 to 6 Hz and the music modulation spectrum has the narrow peak with frequencies below 1 Hz. That difference has been used with Gaussian parameters in the speech and music discrimination method. Lambrou et al [21] have investigated musical signal classification between the three musical genres of rock, piano, and jazz, using different wavelet analysis techniques (Logarithmic splitting, uniform splitting/wavelet packets, and adaptive splitting) in conjunction with statistical pattern recognition methods. Eight statistical measurements were collected from the original signals as

well as from their different wavelet transform coefficients. The statistical features were as follows:

- First Order Statistics in time domain and wavelet transform, that include the mean, variance, skewness, and the kurtosis.
- Second Order Statistics in time domain and wavelet transform, which include angular second moment, correlation, and entropy.
- Number of ZC in time domain and wavelet transform.

Lambrou et al have investigated four different statistical classifiers:

1. Minimum Distance Classifier (MDC),
2. k-Nearest Neighbor Distance Classifier (k-NNC),
3. Least Squares Minimum Distance Classifier (LSMDC), and
4. Quadrature Classifier (QC).

According to the authors, the features selected by the adaptive analysis wavelet transform coefficients performed better than the other techniques. Under the classification rule of either the Minimum Distance Classifier (MDC) or the Least Squares Minimum Distance Classifier (LSMDC). Overall classification accuracy of 91.67% was achieved.

Delfs and Jondral [22] have compared the Discrete Fourier Transform (DFT) and the translation-invariant wavelet packet transform with regard of their use as feature extractors for time-varying transient signals. The authors have used only Piano sounds as a time-varying transient signal and have reported that wavelet packet based methods do not seem to offer any advantages in comparison with simple DFT features. In [23], Ezzaidi and Rouat have addressed the problem of speech, music, and music with songs classification in telephony with handsets variability. Two systems have been investigated; the first system uses three GMMs for speech, music and songs respectively. The second system is based on an empirical transformation of the ΔMFCC, which enhances the dynamical evolution of tonality, and Bayesian classification scheme. They have reported an average of 94.6% classification accuracy.

Hoyt and Wechsler [24] have proposed a method of detecting human speech in the presence of structured noise (e.g. wind, music, traffic sounds, etc.). They have developed two separate algorithms. The first one detects the presence of speech by testing for concave and/or convex formant shapes by using Linear Predictive Coding (LPC) with Discrete Fourier Transform (DFT). In the second algorithm they have used Radial Basis Function (RBF) neural networks with mel-cepstra features vectors. The average accuracy for the first algorithm was around 60% and for second algorithm was around 81%. Mesgarani et al [25] have described a spectro-temporal

auditory method for audio classification into speech and non-speech. In the given method the features are extracted by a biologically inspired auditory model of auditory processing in the cortex. The authors have developed a multi-linear dimensionality reduction algorithm based on Higher Order Singular Value Decomposition (HOSVD) of the multimodal data. They have used Support Vector Machines (SVM) with Radial Basis Function (RBF) to perform classification. They have claimed to achieve 100% classification accuracy for both speech and non-speech classes.

In [26], Jarina et al have proposed an algorithm for speech/music discrimination, which works on data directly taken from MPEG encoded bitstream to avoid the computationally difficult decoding-encoding process. The approach presented in this paper is based on thresholding of features derived from the modulation envelope of the frequency-limited audio signal. The width of the widest peak and average rate of peaks within a time interval of 4 sec are chosen as features for the discriminator. The authors have reported 91% classification accuracy but poor results were obtained for music signals with strong rhythms. To overcome this problem Jarina et al, in [27], introduced a new feature "Rhythm metric" that quantifies the strength of rhythm in audio signals and have incorporated this feature in the model described in [26]. They have reported 97.71% classification accuracy [27].

Nakajima et al [28], have proposed MPEG audio classification algorithm on sub-band

data domain. Classification was performed for silent, speech, music, and applause segments at 1 sec sample. As discriminating features they have used variance of sub-band 0 (zero) energy for silent segment detection, temporal energy distribution, and bandwidth for speech/music discrimination, and center frequency of sub-band for applause detection. After discriminating non-silent segments, MPEG audio stream was classified in speech, music, and applause by using Bayes discriminant function for multivariate Gaussian distribution. Music and speech have been successfully detected at around 90% accuracy.

Li et al [29] have described a procedure which classifies audio signals into seven categories including silence, single speaker speech, multi speakers' speech, music, environmental noise, simultaneous speech and music, and speech and noise. They have investigated a total of 143 features; 68 acoustical features, including eight temporal and spectral features, and 12 features of MFCC, LPC, ΔMFCC, ΔLPC, and autocorrelation MFCC features. These features were extracted every 20 ms from the input data. For each of these 68 features, they have computed the mean and the variance over adjacent frames centered around the frame of interest. Thus, a total of 143 classification features, 68 mean values, 68 variances, pause rate, harmonicity, and five summation features, were computed every 20 ms. The classification is performed using a Bayes classifier. The proposed system provides around 90% accuracy.

In [30], Esmaili et al have performed classification of sounds into 6 music genres consisting of rock, classical, folk, jazz, and pop by using time-frequency analysis. For each 5 sec music segment 10 features are extracted:

1. Mean and standard deviation of centroid frequency,
2. Mean centroid (low-frequency range),
3. Mean of centroid ratio to previous window,
4. Mean bandwidth,
5. Silence ratio,
6. Mean and standard deviation of the frequency location with the lowest energy,
7. Mean and standard deviation of entropy.

Once the features are extracted for the 143 audio signals, linear discriminant analysis (LDA) is then applied using SPSS (Statistical Package for the Social Sciences by SPSS Inc.) software, to predict group classification of cases. 93% classification accuracy is reported.

The next chapter will elaborate more on audio features used for classification and introduce some additional features that we have investigated in this thesis.

Chapter 3

Audio Features-An Overview

In Chapter 2, many features used in audio classification have been mentioned without details. In this chapter, we start by elaborating on audio features that have been commonly used by other researchers for audio classification. In Section 3.2, we will explain details of features that we have considered in audio classification. They have been divided into two parts: the first one explains features that have already been used by others whereas the second gives details of audio features that have been first considered in this thesis, to the best of our knowledge.

3.1 Commonly Used Audio Features

3.1.1 Pitch

Pitch is a perceptual property of periodic or approximately periodic sounds having spectra that contain harmonics of a common fundamental frequency. Pitch should be distinguished from "timbre", which is a perceptual quality relating to the sharpness or dullness of a sound. Timbre is mainly related to spectral shape. Pitch variation conveys intonation, which indicates lexical stress and aspects of syntax. There exists many methods to extract pitch but usually the pitch is estimated by taking a series of short-time Fourier spectrum.

The delta pitch is usually calculated by estimating the trend of the pitch over a particular number of successive frames. For the pitch, signal discrimination is mainly produced by the difference between the mean of the speech and music distribution. The pitch and the delta pitch extracts two different aspects of the signals, their difference in average value and the expected rate of change. Shao et al [31] and Tzanetakis et al [32] have used pitch in their research. In addition to pitch, Carey et al [5] have used delta pitch as well.

3.1.2 Amplitude

Amplitude is the objective measurement of the degree of change in atmospheric pressure caused by sound waves. Typically, it is measured by the *Root-Mean-Square* (RMS) of the audio signal, i.e., squaring the amplitude of each point of a signal and then taking its mathematical average. The signal amplitude, A, is defined as

$$A = \sqrt{\sum_{n=1}^{N} x^2(n)}$$

Strong temporal variations in the amplitude of speech signals are observed due to vowels and vowel-like sounds, fricative consonants, and stop consonants. It appears that music in general has very little amplitude variation between frames when compared with the speech signal. Thus, speech and music are distinguished by the distribution of amplitude values. The delta amplitude is calculated by estimating the trend of the amplitude over some successive frames. Carey et al [5] have used both amplitude and delta amplitude. In [13, 17, 32] RMS has been used for classification between speech and music.

3.1.3 Zero-Crossing Rate

The zero-crossing rate is a measure of the number of times in a given time interval that the audio signal amplitude passes through a value of zero. The zero-crossing rate corresponding to the short low energy consonants is lower than the zero-crossing rate corresponding to the longer higher energy vowels. According to Saunders [2], the

zero-crossing rate provides a measure of the weighted average of the spectral energy distribution in the waveform, which is referred to as the spectral center of the mass. The zero-crossing rate is dominated by the strongest spectral component in the signal. Kedem [33] calls it a measure of the dominant frequency in a signal, which is a correlate of the spectral centroid. Many researchers have investigate zero-crossing rate for speech/music classification [2, 3, 4, 5, 17, 32, 34, 35, 36, 37, 38, 39]. Carey et al [5] have also used delta zero-crossing rate which is calculated by estimating the trend of the zero-crossing rate over some successive frames. El-Maleh et al [16] have used the number of zero-crossing of the filtered input signal. Lambrou et al [21] have also used the number of zero-crossings in both time domain and wavelet domain.

3.1.4 Silence Crossing Rate

In [12], Harb and Chen have proposed this feature. According to them there exist an alternation in the energy between peaks, corresponding to words, and almost zero energy, corresponding to inter-words' silence. This characteristic is helpful for detecting speech signals. So, they proposed it to count in a window of 1 second the number of times the energy falls below the silence level, and named it the Silence Crossing Rate (SCR). They have shown that the SCR is in the range of 5 to 10 for speech. For other types of audio, such as music, it is either higher or lower.

3.1.5 Spectral Roll-Off Point

This value measures the frequency below which 95% of the power in the spectrum resides. This is a measure of the "skewness" of the spectral shape. This value is higher for right-skewed distributions [4]. Voiced speech has a high proportion of energy contained in the low frequency range of the spectrum, whereas most of the energy for unvoiced speech and music is contained in higer bands. As a result the spectral roll-off point exhibits higher values for music and unvoiced speech, and lower values for voiced speech. The spectral roll-off value for a frame is computed as follows:

$$\sum_{f<K} X[f] = 0.95 \sum_{f} X[f]$$

where K is the spectral roll-off point and $X[f]$ is the power of the signal at the corresponding frequency f. Other researchers have also used this feature in speech/music classification [3, 4, 35, 36, 37, 38].

3.1.6 Spectral Centroid

A number of researchers have been using a definition for the "spectral centroid" as a physical measure closely correlating with perceptual "brightness". It represents the balancing point of the spectral power distribution within a frame. Many kinds of music involve percussive sounds which, by including high-frequency noise, push the spectral mean higher. In addition, excitation energies can be higher for music

than for speech where pitch stays in a fairly low range. This measure gives different results for voiced and unvoiced speech [3, 4]. The spectral centroid for a frame occurring at time t is computed as follows [3]:

$$\text{Spectral Centroid} = SC = \frac{\sum_{k=1}^{N-1} kX(k)}{\sum_{k=1}^{N-1} X(k)} \tag{3.1}$$

where index k is a small band of frequencies within the overall measured spectrum, $X(k)$ is the magnitude of the signal at the corresponding frequency band, and N is the length of the Discrete Fourier Transform (DFT). This feature has been used in [3, 4, 30, 31, 32, 34, 35, 36, 37, 38, 40].

3.1.7 Bandwidth of a Signal

This feature shows the spectral shape and the spread of energy relative to the centroid. For example, a single sine wave has a bandwidth of zero and an ideal white noise has an infinite bandwidth [31]. Speech usually has a narrower bandwidth than music. Bandwidth is computed as the magnitude-weighted average of the differences between the spectral components and the centroid. It is defined as:

$$\text{Bandwidth} = \sqrt{\frac{\sum_k (SC - k)^2 X(k)}{\sum_k X(k)}}$$

where SC is the spectral centroid given in equation 3.1. Many researchers [30, 31, 34, 35, 36, 41] have used this feature in speech/music classification.

3.1.8 Pulse Metric

Scheirer et al [4] have proposed this feature which was intended to correlate with the perceptual sense of a strong rhythmic beat in the signal. The beat of a piece of music is one of the clearest features of the music to both musicians and non-musicians. If a regular beat is found in a signal, it is almost certainly a musical signal. According to the authors, this method is useful to detect a strong driving beat in music, but fails when the rhythm deviates very much from a central time, as in rubato[1] music.

3.1.9 4 Hz Modulation Energy

Speech signal has a characteristic energy modulation peak around the 4 Hz syllabic rate. In order to model this property, the signal is segmented in 16 ms frames. Mel Frequency Spectrum Coefficients are extracted and energy is computed in 40 perceptual channels. This energy is then filtered with Finite Impulse Response (FIR) band pass filter, centered on 4 Hz. Energy is summed for all channels, and normalized by the mean energy of the frame. The modulation is obtained by computing the variance of filtered energy in dB for every one second of the signal. Speech carries more modulation energy than music [4, 7, 8, 9, 10].

[1] Italian technique in music, subtle rhythmic manipulation and nuance in performance for greater musical expression.

3.1.10 4 Hz Harmonic Coefficients

Chou et al [10] have proposed a new feature called Harmonic Coefficients to represent the characteristics of harmonic structures of voiced speech, which is calculated by the average maximum autocorrelation value in time-domain and frequency-domain. They have used this feature for singing detection, and according to the authors, the accuracy of singing detection can be enhanced by 4 Hz modulation harmonic coefficients. Similar to 4-Hz modulation energy, speech signals have a characteristic harmonic feature modulation peak around 4-Hz syllabic rate, as the signals periodically change between voiced speech and unvoiced speech. Singing segments usually have long duration of consonants with low 4 Hz modulation value. According to the authors, 4 Hz modulation value of harmonic coefficient can be used as a complementary feature with full-band harmonic coefficient and 4-Hz modulation energy in singing detection.

3.1.11 Energy Contour

Saunders has mentioned in [2] that the energy contour of a waveform is capable of separating speech from music. The contour of music tends to show a much smaller number of dips and peaks than speech and it quite often shows little change over a period of several seconds. The alternation between voicing and frication in speech produces a marked change in its energy coutour.

3.1.12 Entropy Modulation

Music appears to be more "ordered" than speech when observations are made of both signals and spectrograms. To measure the "disorder" of speech, Pinquier et al [7, 8, 9] have evaluated a feature based on signal entropy. The signal is segmented in 16 ms frames. The entropy is then computed on every frame. This measure is used to compute the modulation of entropy on one second of the signal. The modulation of entropy is higher for speech than it is for music.

3.1.13 Frequency Tracking

Harb et al [12] have proposed this audio feature for speech/music classification. In the spectrum of the signal, the authors have tried to track the five most important frequencies in the Discrete Fourier Transform (DFT) vectors. In other words, they observe whether these frequencies continue to be the most important frequencies within 100 Hz band in the next DFT vector. When a frequency cannot been tracked, they signal a cut. The number of cuts in a window of 1 sec is the Frequency Tracking feature. The frequency tracking for speech signals gives short dispersed segments and is higher than that for music. For music, the changes in the spectrum are smooth in general, so the frequency tracking gives longer parallel segments.

3.1.14 Modulation Spectrum

Speech and music modulation spectrum for a given sub-band have been used and analyzed by Pavel Balabko in [20] for the rhythmical properties of the signal. The rhythmical properties of the signal are quite different for speech and music. Speech modulation spectrum has a typical wide peak at frequencies from 2 to 6 Hz and the music modulation spectrum has the narrow peak with frequencies below 1 Hz. This difference is caused by different energy changing for speech and music data. According to the author, the typical rate of speech energy changes corresponds to the average syllable rate (around 4 Hz) and the rate of music energy changes corresponds to the beat rate (around 0.7 Hz).

3.1.15 Cepstral Coefficients

The cepstral coefficient are used to describe the short-term spectral envelop of a speech signal. The cepstrum is the inverse Fourier transform of the logarithm of the short-term power spectrum of the signal, which is given by

$$c(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} log\,|X(\omega)|\, e^{j\omega n} d\omega \tag{3.2}$$

Cepstral analysis attempts to separate source from filter, hence it can be viewed as deconvolution. Only a moderate number of ceptral coefficients ranging between 10 and 14 are needed for most applications, including speech analysis. In Speech

Processing, the cepstral coefficients are used to obtain the formants from voiced phonemes. The information relevant to the formants is contained in the first coefficients of the cepstrum. This feature has been examined in [36, 39].

3.1.16 Cepstral Residual

Scheirer et al in [4] showed that if real cepstral analysis and smoothing of the spectrum is carried out, followed by re-synthesizing and comparing the smoothed to un-smoothed spectrum, a better fit for unvoiced speech is obtained than that for voiced speech or music. This is because unvoiced speech better fits the homomorphic single source-filter model than music does. In the voiced speech, the authors filter out the pitch "ripple" from the signal, giving higher values for the residual.

3.2 Audio Features Used in This Work

In this section we will discuss those features that we have used for speech/music classification. We have also proposed using audio features for classification purposes for the first time, according to our knowledge, and have utilized them with some other previously used audio features. The next subsection will discuss previously used audio features whereas section 3.2.2 will present our newly proposed audio features for music/speech classification.

3.2.1 Previously Used Audio Features

Among the many features that have been used by other researchers for music/speech classification, we considered three features, namely the percentage of low energy frames, the spectral flux, and the linear predictive coefficients.

Percentage of "Low Energy" Frames

This value measures the proportion of frames with root mean-squared (RMS) power less than 50% of the mean RMS power within a given period of time. According to [2] the energy distribution for speech is more left-skewed than that of music. The reason is that there are more quiet frames in speech as some pause between every word exists and hence the energy of the frame containing pauses is lower than frames containing no pauses. This measure will be higher for speech than that for music as shown in Figure 3.1. This feature has also been used by many researchers [3, 4, 31, 35, 36, 37, 38, 40].

Spectral Flux

This feature, also known as the Delta Spectrum Magnitude measures frame-to-frame spectral difference. Thus, it characterizes the changes in the shape of the spectrum. Speech goes through more drastic frame-to-frame changes than music. Speech alternates between periods of transition (consonant-vowel boundaries) and periods of relative stasis (vowels), whereas music typically has a more constant

rate of change. As a result the spectral flux value is higher for speech than it is for music as shown in Figure 3.2 [34, 39, 42]. However, Scheirer et al in [4] have reported that this value is higher for music than it is for speech, which is theoretically not possible as there are more frame-to-frame changes in speech than in music. According to [34], this feature is especially useful for discriminating some strong periodicity environment sounds such as tone signal, from music signals. It is defined as the 2-norm of the frame-to-frame spectral amplitude difference vector, and is given by:

$$\text{Spectral Flux} = \| \, |X_i| - |X_{i+1}| \, \| \tag{3.3}$$

In the literature review, we found that many researchers have used this feature [3, 4, 34, 35, 36, 37, 38, 39, 42, 43].

Linear Predictive Coefficients (LPC)

The basic idea behind linear prediction is that the next signal sample is predicted from a weighted sum of p previous samples, given as follows:

$$\hat{s}(n) = \sum_{i=1}^{p} a_i s(n-i) \tag{3.4}$$

where a_i represent the prediction coefficients, p is the predictor order, and $s(n-i)$ is a sample at time instant $n-i$. In other words, each sample of a signal is modeled as a linear combination of previous samples. The prediction coefficients are determined by minimizing the mean squared error between the actual sample and the prediction.

A reasonable number of coefficients for speech analysis is between 10 and 20. The prediction error signal, also called residual error, is given by:

$$e(n) = s(n) - \hat{s}(n) = s(n) - \sum_{i=1}^{p} a_i s(n-i) \tag{3.5}$$

Prediction error is significantly higher for unvoiced speech than it is for voiced speech. The linear prediction coefficients that we have used in our work uses the Levinson-Durbin recursion to solve the normal equations that arise from the least-squares formulation. This computation of the linear prediction coefficients is often referred to as the autocorrelation method. Many researchers have used linear prediction coefficients in their music/speech classification research [18, 24, 29, 32, 36].

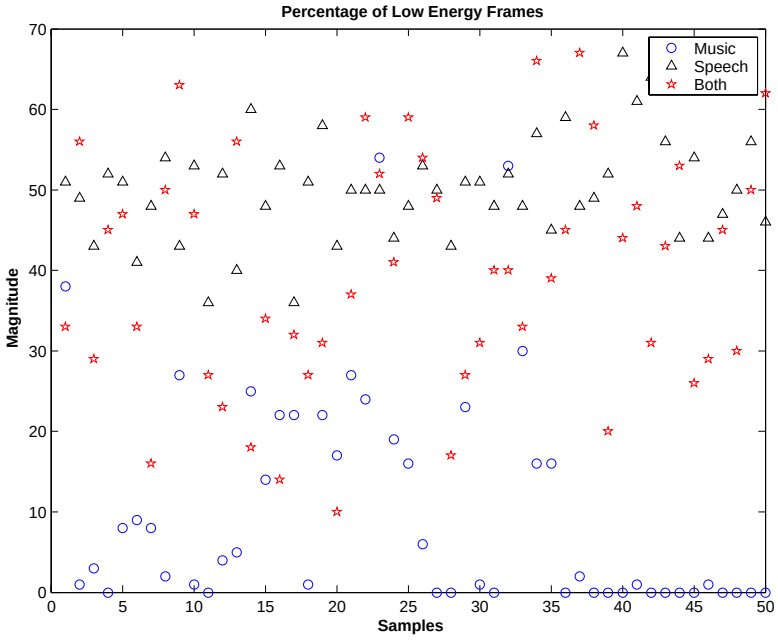

Figure 3.1: Percentage of "Low Energy Frames"

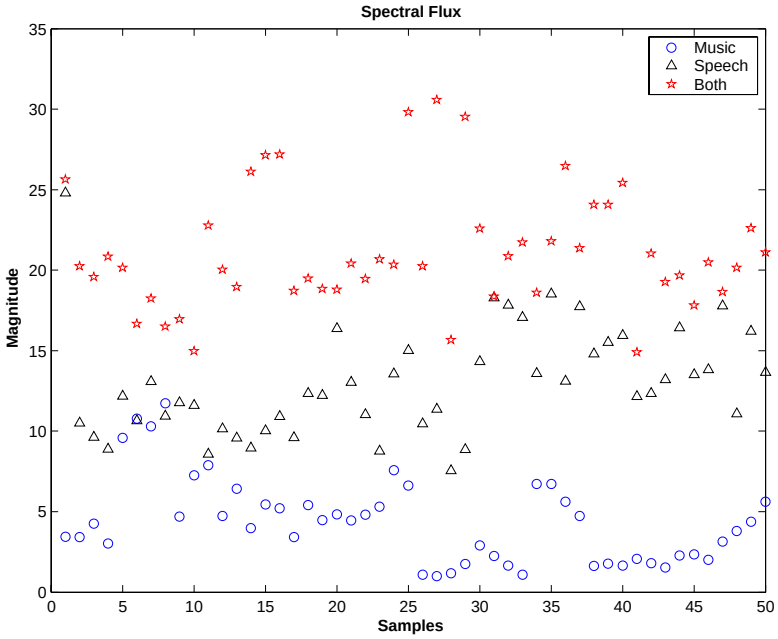

Figure 3.2: Spectral Flux

3.2.2 Newly Proposed Audio Features

In this section we will present audio features that we were the first to propose and study for the purpose of music/speech classification.

Range of Zero-Crossings

As discussed in Section 3.1.3, zero-crossings is a measure of the number of times in a given time interval that the audio signal amplitude passes through a value of zero. Rather than using zero-crossings directly, we have used the range of zero-crossings as a feature vector, since speech goes through more drastic frame-to-frame changes

than music. That is, speech alternates between periods of transition (consonant-vowel boundaries) and periods of relative stasis (vowels), whereas music typically has a more constant rate of change. As a result the range of zero-crossings for speech is higher than for music. It is evident from Figure 3.3 that it gives discriminating patterns for different classes of audio signal.

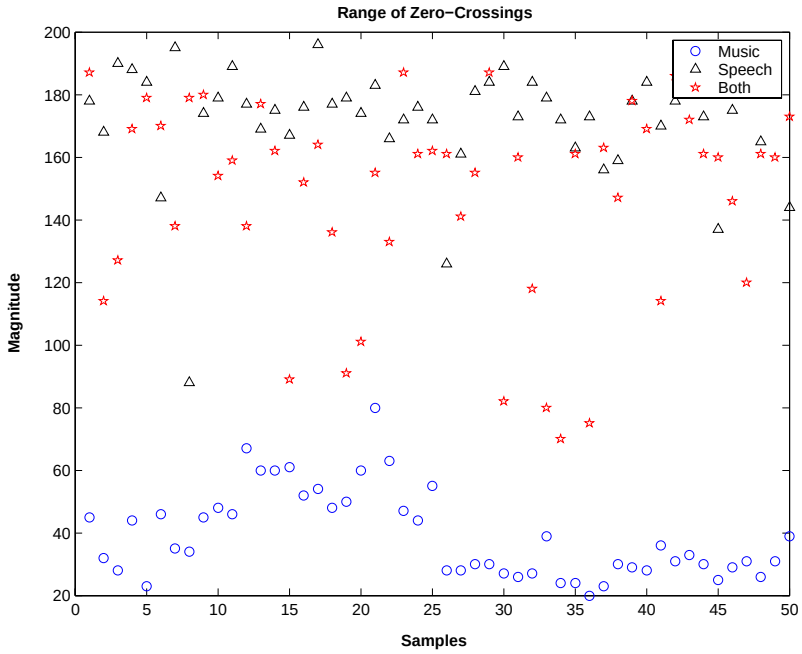

Figure 3.3: Range of Zero-Crossings

Discrete Wavelet Transform

A serious drawback for using Fourier transform is that after transforming the audio signal into the frequency domain, the time information is lost. When looking at a Fourier transform of a signal, it is impossible to tell when a particular event

took place. The Discrete Wavelet Transform maps a signal into a two-dimensional function of time and frequency. Wavelet analysis is capable of revealing aspects of data that other signal analysis techniques miss, which include trends, breakdown points, discontinuities in higher derivatives, and self-similarity. In wavelet analysis, a signal is split into an *approximation* and a *detail*. The approximation is then itself split into a second-level approximation and detail, and the process is repeated. For an n-level decomposition, there are $n + 1$ possible ways to decompose or encode the signal. The approximations are the low-frequency components of the signal, whereas the details are the high-frequency components. Since we have only single dimensional data, we have used a single-level, 1-D 'Haar' wavelet transformation. We have investigated the statistical features of audio in the wavelet domain which are the mean, shown in Figure 3.4 and the variance, shown in Figure 3.5. Lambrou et al [21] have also used these features but only for music genre classification and not for speech/music classification. Delfs et al [22] have used wavelet packet transform for classification of piano sound. In wavelet packet analysis, the details as well as the approximations can be split into n-levels. This yields $2n$ different ways to encode the signal.

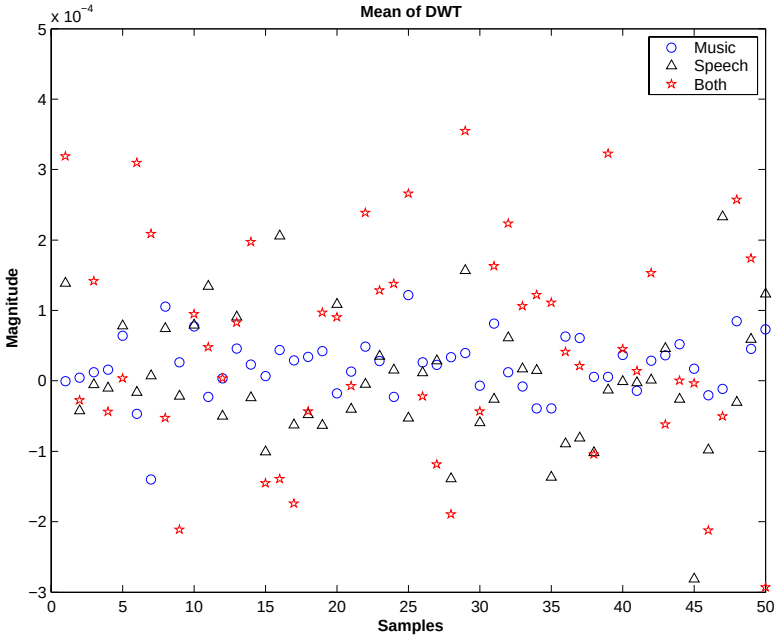

Figure 3.4: Mean of Discrete Wavelet Transform

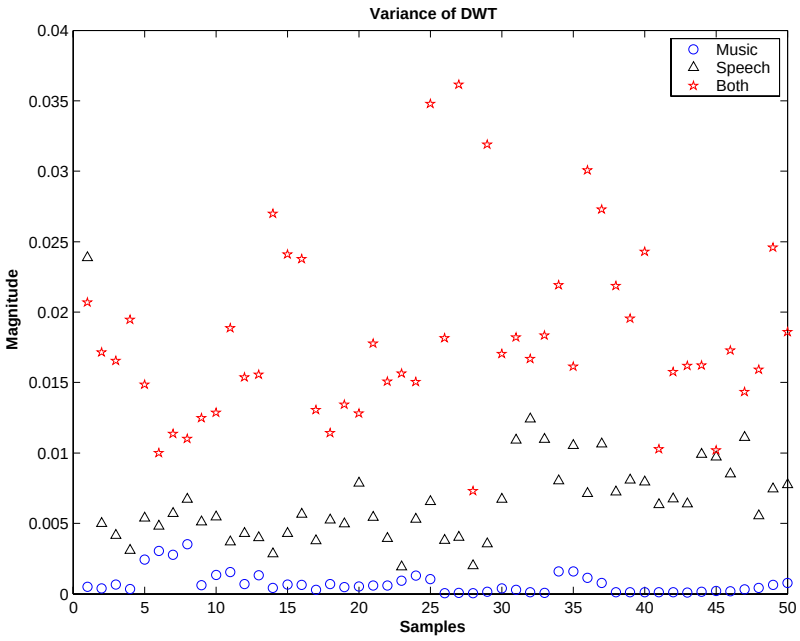

Figure 3.5: Variance of Discrete Wavelet Transform

RMS of a Lowpass Signal

Music signals have a wider bandwidth than speech extending up to 20 kHz. To limit the frequency band we have applied a lowpass filter to filter out the high frequency contents. We have applied Butterworth filter of 4^{th} order with 1.1 kHz cutoff frequency. After that, we have taken a Root Mean Square value of that lowpass response. The RMS value of a lowpass response for speech is higher than the RMS value of a low-pass response for music because most of the speech contents are in the lower frequency band as shown in Figure 3.6.

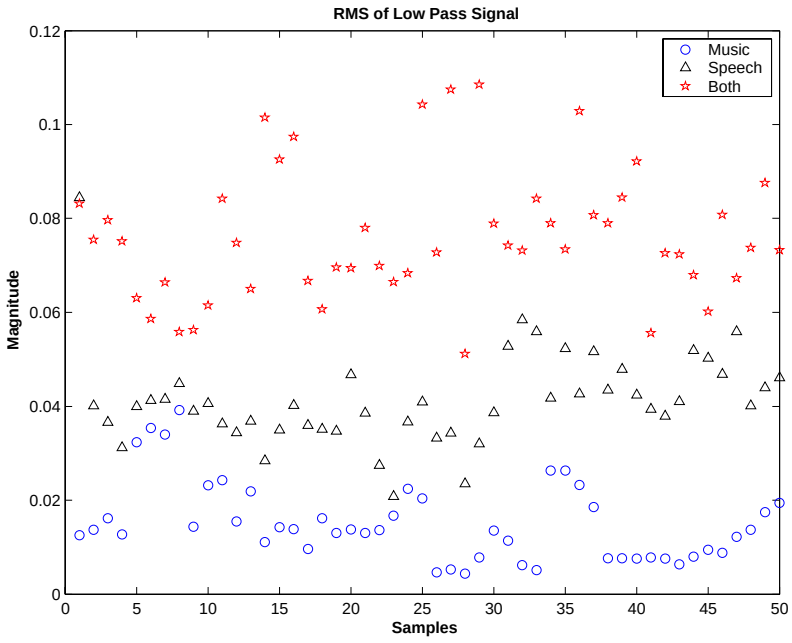

Figure 3.6: RMS of a Low-Pass Signal

Mel Frequency Cepstral Coefficients (MFCC)

The main purpose of the MFCC is to imitate the behavior of a human ear. Psychophysical studies have shown that human perception of the frequency contents of sounds for speech signals does not follow a linear scale. Thus for each tone with an actual frequency f, measured in Hz, a subjective pitch is measured on a scale called the *Mel* scale. The Mel-frequency scale is a linear frequency spacing below 1 kHz and a logarithmic spacing above 1 kHz. Therefore, filters spaced linearly at low frequency and logarithmic at high frequencies can be used to capture the phonetically important characteristics (voiced and unvoiced) of the speech. The commonly used formula to approximately reflect the relation between the Mel-frequency and the physical frequency is given by:

$$M(f) = 1125 * \log_{10}(1 + \frac{f}{700}) \tag{3.6}$$

Mel-frequency cepstral coefficients (MFCC) are perceptually motivated features that are also based on the STFT (Short Time Fourier Transform). After taking the log-amplitude of the magnitude spectrum, the FFT (Fast Fourier Transform) bins are grouped and smoothed according to the perceptually motivated Mel-frequency scaling. Finally, in order to decorrelate the resulting feature vectors, a Discrete Cosine Transform (DCT) is performed. Although typically 12 coefficients are used for speech representation, [35, 40] have found that the first five coefficients provide the best classification performance. [7, 10, 12, 14, 20, 23, 29, 31, 32, 34, 36, 37, 38, 44]

have used MFCC in their research and [23, 29, 44] have used derivative of MFCC (Δ MFCC). We have investigated the variance of MFCC in this thesis.

Chapter 4

Selection of Classification Features

The first step in a classification problem is typically data reduction. The data reduction stage which is also called feature extraction, consists of discovering a few important facts about each class. Since audio data contains much redundancy, important features are lost in the dissonance of unreduced data. The choice of features is critical as it greatly affects the accuracy of audio classification. The selected features must reflect the significant characteristics of each class of audio signals. In order to better discriminate different classes of audio, we consider features that are related to temporal and spectral domains.

Typically, audio features are extracted in two levels: short-term frame-level and long-term clip-level. Here, a frame is defined as a group of adjacent samples lasting for 10 to 40 ms. The audio signal within such periods presumably remains sta-

tionary and short-term features both in time-domain and frequency-domain can be extracted. For a feature to reveal the semantic meaning of an audio signal, we need to observe the temporal variation of frame features on a longer time scale, usually from one second to several tens of seconds. Such an interval is called an audio clip. An audio clip consists of a sequence of frames and clip-level features that characterize how frame level features change over a clip. The clip boundaries may be the result of audio segmentation such that the content within each clip belongs to the same class. Fixed length clips, lasting for 2 to 3 sec may also be used in determining clip boundaries.

In this chapter we present the features that we have studied in our research. They include: *Root Mean Square of Lowpass Signal (RMS-LPS)*, *Mean of Discrete Wavelet Transform (M-DWT)*, *Variance of DWT (V-DWT)*, *Spectral Flux (SF)*, *Percentage of Low Energy Frames (%LEF)*, *Range of Zero Crossings (R-ZC)*, *Linear Prediction Coefficients (LPC)*, and *Variance of Mel Frequency Cepstral Coefficients (V-12MFCC)*. The details of each feature has already been discussed in Chapter 3.

We have extracted RMS-LPS, M-DWT, V-DWT, SF, %LEF, and R-ZC at frame-level where each frame is of 20 ms duration. Each clip was of 3 sec duration containing 150 frames in each clip. After extracting those features at frame-level we have taken the mean of 150 values of each feature to get a single feature value for each

clip. For instance, we will get 150 values of M-DWT and V-DWT each belonging to a single frame. After that we will take the mean of 150 M-DWTs and the mean of 150 V-DWTs to get single M-DWT and V-DWT feature vector for each clip. In case of R-ZC, we have first calculated the number of zero-crossings in each frame and then we subtracted the minimum zero-crossings from the maximum zero-crossings within a clip to get R-ZC. The 12 coefficients of both LPC and V-12MFCC were extracted at clip-level, where each clip was of 3 sec duration. The feature vector of each clip consists of the following: a single value of RMS-LPS, M-DWT, V-DWT, SF, %LEF and R-ZC, 12 coefficients of LPC, and 12 coefficients of MFCC.

In order to verify the "discriminating abilities" of each feature, researchers have used different techniques such as cluster analysis, distance measures, entropy analysis, and other related methods [18]. In this research, we have employed cluster analysis in order to assist us in selecting the best combination of features that gives the highest classification accuracy. In particular, we have used the "*Fuzzy C-Mean Clustering*" as discussed in the rest section.

4.1 Fuzzy C-Mean Clustering

Clustering of numerical data forms the basis of many classification and system modeling algorithms. The purpose of clustering is to identify natural groupings of data

from a large data set to produce a concise representation of a system's behavior.

In the conventional K-Means algorithm, each data point is assumed to be the member of exactly one cluster. In pattern classification domain, such memberships are rarely seen and a classifier bearing a feature vector of more than 2 dimensions is normally considered to be partially associated to one specific domain. This partial association can be efficiently described and presented on the basis of a Fuzzy version of K-Means Clustering algorithm known as *Fuzzy C-Means Algorithm*. Fuzzy C-Means is a data clustering technique wherein each data point belongs to a cluster to some degree that is specified by a membership grade. This technique was originally introduced by James C. Bezdek in [45] as an improvement on earlier clustering methods. It provides a method that shows how to group data points that populate some multidimensional space into a specific number of different clusters [46].

We have applied the Fuzzy C-Mean clustering algorithm to find the contribution of each feature in classification of audio into three different classes: *Music*, *Speech*, and *Speech+Music* (i.e. Speech with background Music). We have then applied the algorithm on all possible combinations of those features in order to determine the best combination of features that would achieve the highest classification accuracy. Since we are only examining the contribution of each feature, we have selected few audio samples from our audio database to extract those features. We have selected

50 audio samples of each class i.e. 50 samples of Music, 50 samples of Speech, and 50 samples of Speech+Music. In the samples of both Speech and Speech+Music, the language was "English" and the speaker was "Male".

4.2 Individual Feature Contribution to Classification

We have taken into account the percentage accuracy of each class to be above 80% in order to consider the feature's contribution for that class as significant. Tables 4.1, 4.2, and 4.3 and Figures 4.1, 4.2, and 4.3, respectively, show that RMS-LPS, SF, and V-12MFCC are good features for classification of all three classes. It is obvious that the data samples cannot be perfect, i.e. there lie some ambiguities among the samples belonging to the same class. For example, in the samples of Speech+Music we may have varying volume of background music making speech dominant or music dominant explaining some of those ambiguities. Similarly, in case of Speech, we may have some noise in the background, that could be mistaken by the classifier as background music.

Table 4.1: Clustering results for RMS-LPS

RMS of Lowpass Signal			
Cluster	***Classes***		
	Music	**Speech**	**Speech+Music**
Music	92%	6%	0%
Speech	8%	92%	12%
Speech+Music	0%	2%	88%
Total	100%	100%	100%

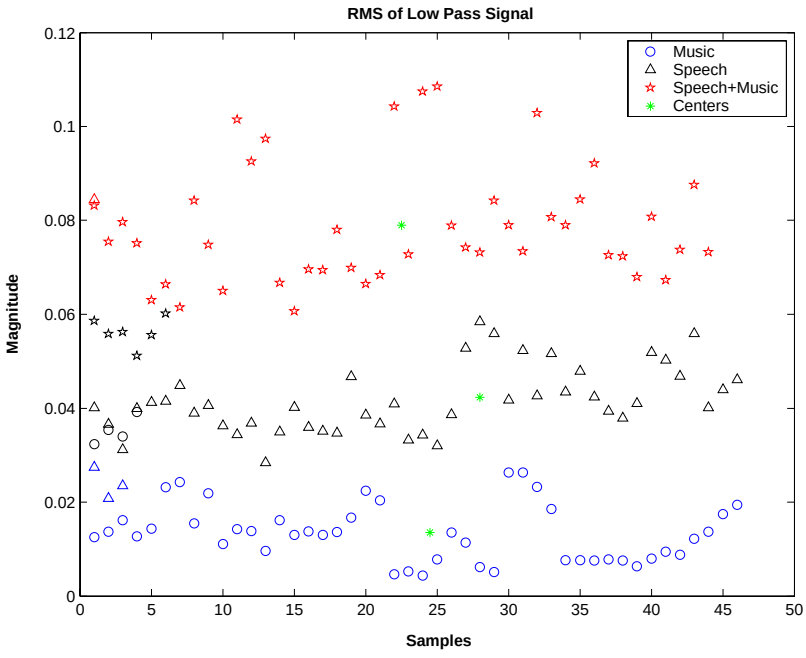

Figure 4.1: Clusters for RMS-LPS

Table 4.2: Clustering results for SF

Spectral Flux			
Cluster	***Classes***		
	Music	**Speech**	**Speech+Music**
Music	92%	2%	0%
Speech	8%	84%	10%
Speech+Music	0%	14%	90%
Total	100%	100%	100%

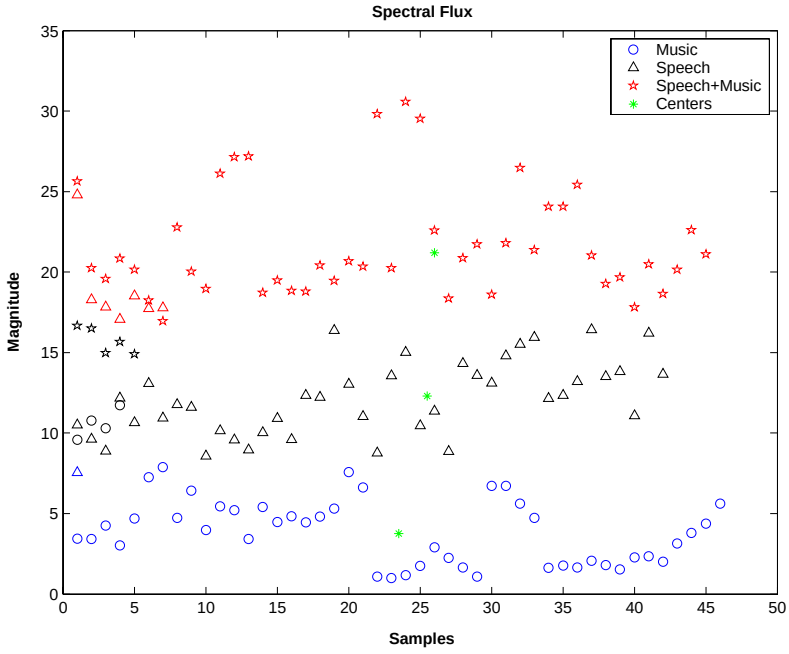

Figure 4.2: Clusters for SF

Table 4.3: Clustering results for V-12MFCC

Variance of MFCC (12 Coeff.)			
Cluster	***Classes***		
	Music	**Speech**	**Speech+Music**
Music	98%	0%	6%
Speech	0%	86%	2%
Speech+Music	2%	14%	92%
Total	100%	100%	100%

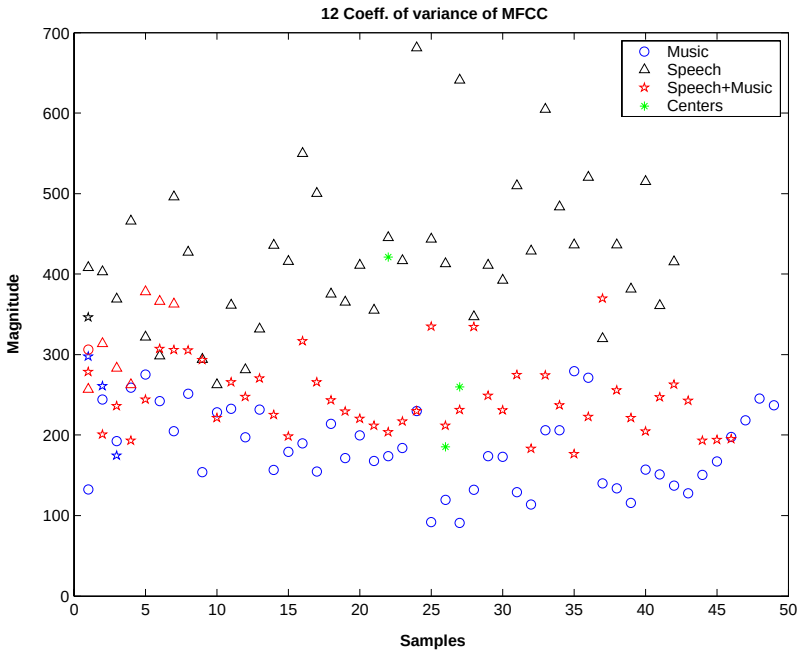

Figure 4.3: Clusters for V-12MFCC

While investigating the variance of MFCC, we have applied the Fuzzy C-Mean clustering algorithm on each of the 12 coefficient as shown in Figure 4.3 and different combinations of those coefficients, as shown in Table 4.5. We have found that the first 4 coefficients, as shown in Figure 4.4, give the same results when using the 12 coefficients as shown in Table 4.4. Furthermore, we applied the same experiment while increasing the number of coefficients but the clustering accuracy remained the same. Therefore, we may conclude that the first 4 MFCCs (V-4MFCC) are enough for audio classification. This is in conformance to what [35, 40] have reported of getting the best classification performance by including the first five coefficients only.

Table 4.4: Clustering results for V-4MFCC

Variance of MFCC (4 Coeff.)			
Cluster	***Classes***		
	Music	**Speech**	**Speech+Music**
Music	98%	0%	6%
Speech	0%	86%	2%
Speech+Music	2%	14%	92%
Total	100%	100%	100%

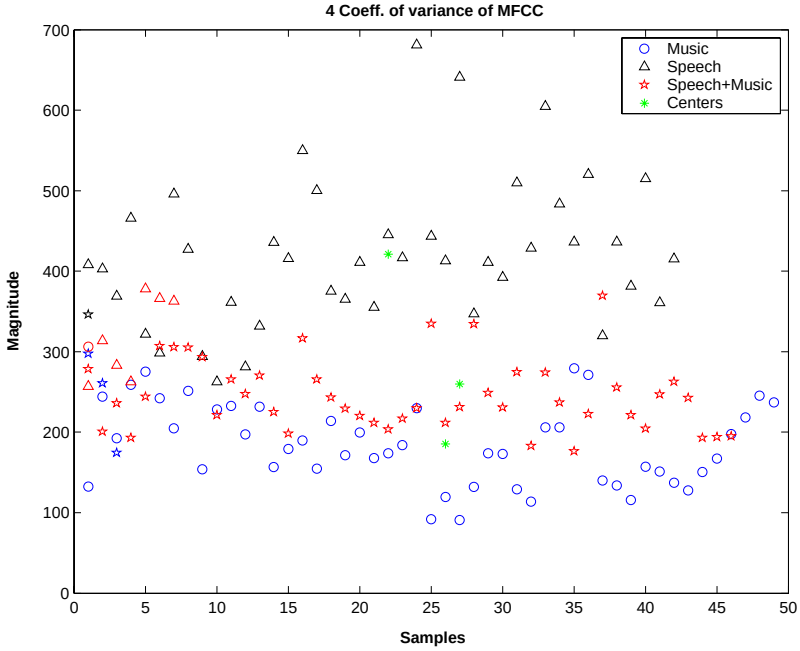

Figure 4.4: Clusters for V-4MFCC

Regarding the use of wavelets as features, there exist many families of wavelets, like 'Haar wavelet', 'Daubechies wavelets', 'Meyer wavelet', 'Mexican hat wavelet', etc.. We have investigated Haar wavelets, Meyer wavelets, and two types of Daubechies wavelets DB2 and DB15. Figures 4.5, 4.6, 4.7, and 4.8 show that the features extracted when using Meyer or DB15 wavelets do not contribute much in the process of classification.

Table 4.5: Clustering results for each coefficient of variance of MFCC

Variance of MFCC			
Coefficient #	***Cluster***		
	Music	**Speech**	**Speech+Music**
1	M=80% S=0% SM=46%	M=0% S=60% SM=0%	M=20% S=40% SM=54%
2	M=96% S=0% SM=4%	M=0% S=72% SM=10%	M=4% S=28% SM=86%
3	M=92% S=0% SM=22%	M=0% S=62% SM=0%	M=8% S=38% SM=78%
4	M=92% S=0% SM=28%	M=0% S=76% SM=8%	M=8% S=24% SM=64%
5	M=98% S=2% SM=4%	M=0% S=38% SM=24%	M=2% S=60% SM=72%
6	M=90% S=0% SM=10%	M=0% S=70% SM=8%	M=10% S=30% SM=82%
7	M=98% S=2% SM=4%	M=0% S=64% SM=26%	M=2% S=34% SM=70%
8	M=100% S=0% SM=10%	M=0% S=58% SM=24%	M=0% S=42% SM=66%
9	M=96% S=0% SM=4%	M=0% S=46% SM=18%	M=4% S=54% SM=78%
10	M=100% S=2% SM=28%	M=0% S=46% SM=2%	M=0% S=52% SM=70%
11	M=94% S=4% SM=26%	M=0% S=68% SM=14%	M=6% S=28% SM=60%
12	M=94% S=0% SM=12%	M=0% S=52% SM=24%	M=6% S=48% SM=64%

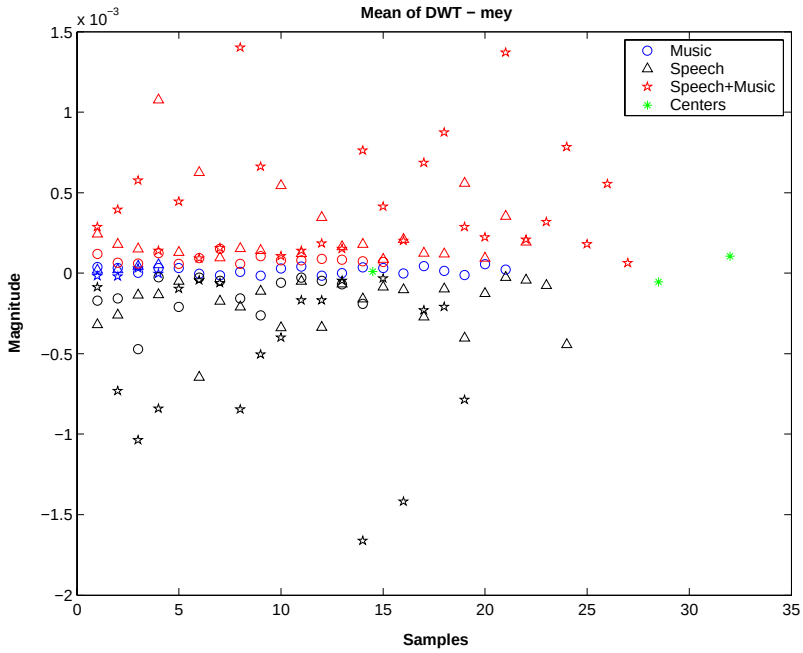

Figure 4.5: Clusters for Mean of DWT "Meyer"

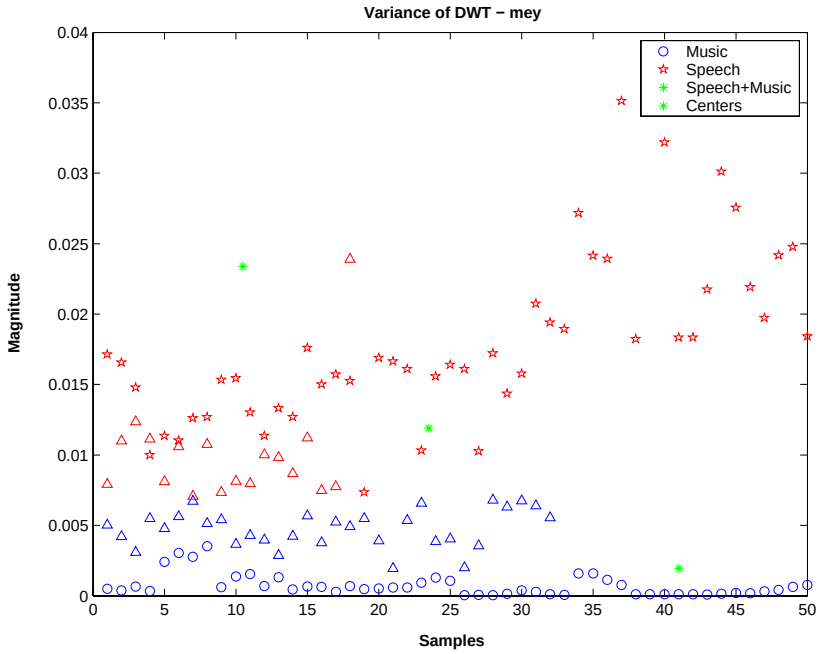

Figure 4.6: Clusters for Variance of DWT "Meyer"

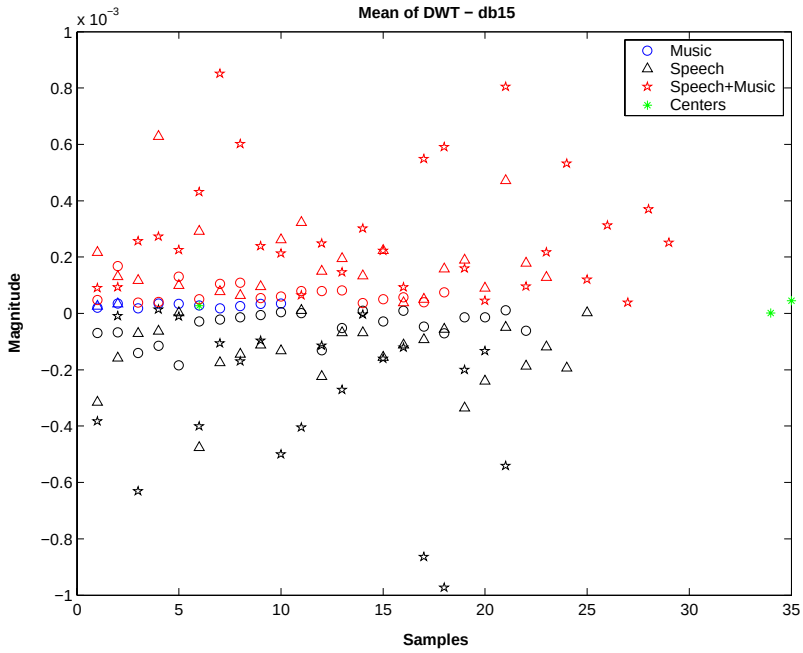

Figure 4.7: Clusters for Mean of DWT "DB15"

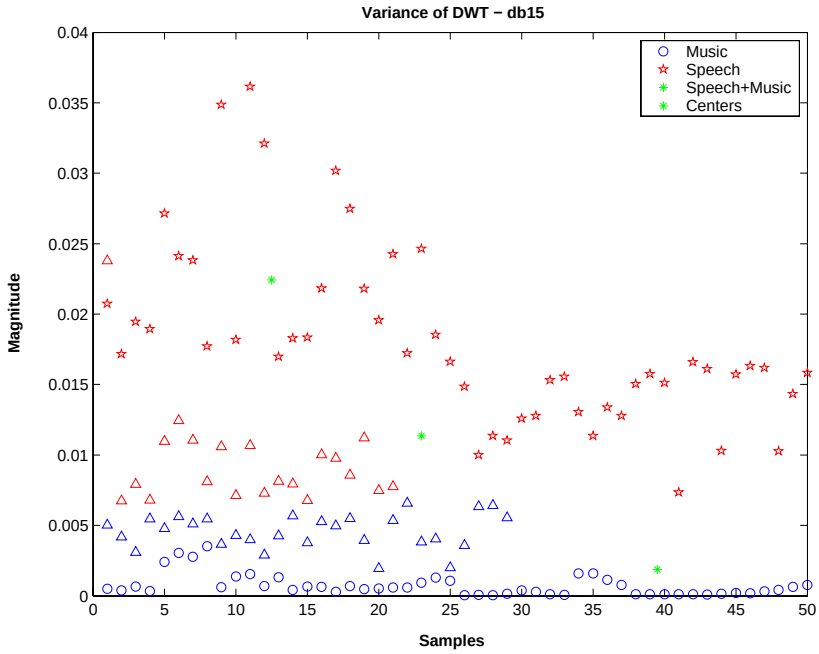

Figure 4.8: Clusters for Variance of DWT "DB15"

The results for the Haar wavelets, as shown in Figures 4.9 and 4.10, indicate that they performed better clustering than DB2 wavelets, shown in Figures 4.11 and 4.12. This is further emphasized in Tables 4.6 and 4.8 showing the use of the mean, and Tables 4.7 and 4.9 showing the use of the variance. Hence, our clustering investigation has directed us to include Haar wavelets and discard the rest. From this point on, when we say wavelet transform, we mean Haar wavelet transform.

According to Figure 4.9, Tables 4.6 and 4.12, M-DWT and LPC do not participate that much in the process of audio classification. However, it could be possible that they may be useful when used with other features. Figure 4.10 and Table 4.7 clearly show that V-DWT is a good feature to classify Music and can be useful for Speech+Music as well. Figure 4.13 and Table 4.10 show that %LEF is a good feature to classify Speech data.

Figure 4.14 and Table 4.11 show that the R-ZC can be useful to classify Music and Speech data but not Speech+Music data. It is worth mentioning that it is not necessary for a given feature to be able to classify audio data for all categories efficiently. Hence, we look for a set of features that gives the highest classification accuracy as a whole.

Table 4.6: Clustering results for Mean of DWT "Haar"

Mean of Discrete Wavelet Transform			
Cluster	***Classes***		
	Music	**Speech**	**Speech+Music**
Music	32%	14%	4%
Speech	36%	58%	44%
Speech+Music	32%	28%	52%
Total	100%	100%	100%

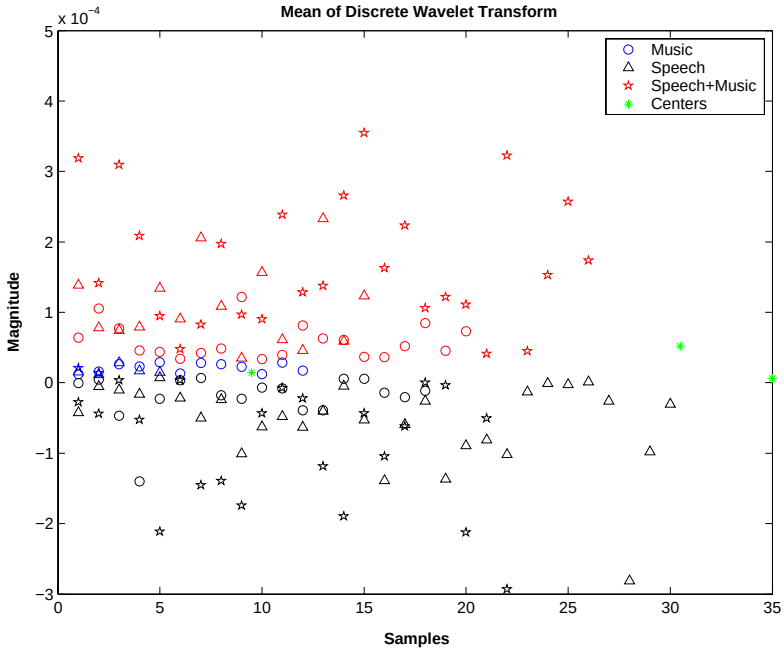

Figure 4.9: Clusters for Mean of DWT "Haar"

Table 4.7: Clustering results for Variance of DWT "Haar"

Variance of Discrete Wavelet Transform			
Cluster	***Classes***		
	Music	**Speech**	**Speech+Music**
Music	100%	48%	0%
Speech	0%	50%	32%
Speech+Music	0%	2%	68%
Total	100%	100%	100%

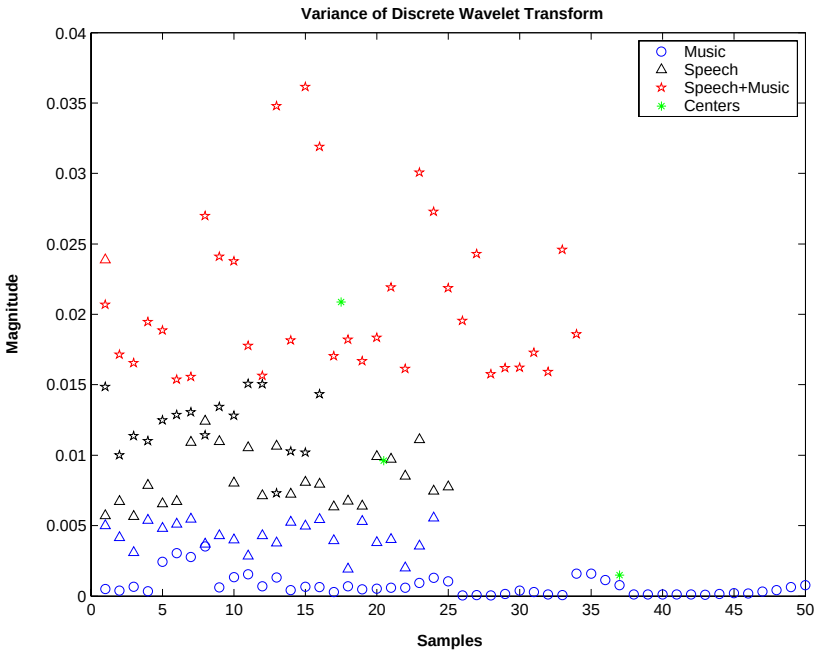

Figure 4.10: Clusters for Variance of DWT "Haar"

Table 4.8: Clustering results for Mean of DWT "DB2"

Mean of Discrete Wavelet Transform			
Cluster	***Classes***		
	Music	**Speech**	**Speech+Music**
Music	18%	8%	4%
Speech	41%	64%	40%
Speech+Music	40%	28%	56%
Total	100%	100%	100%

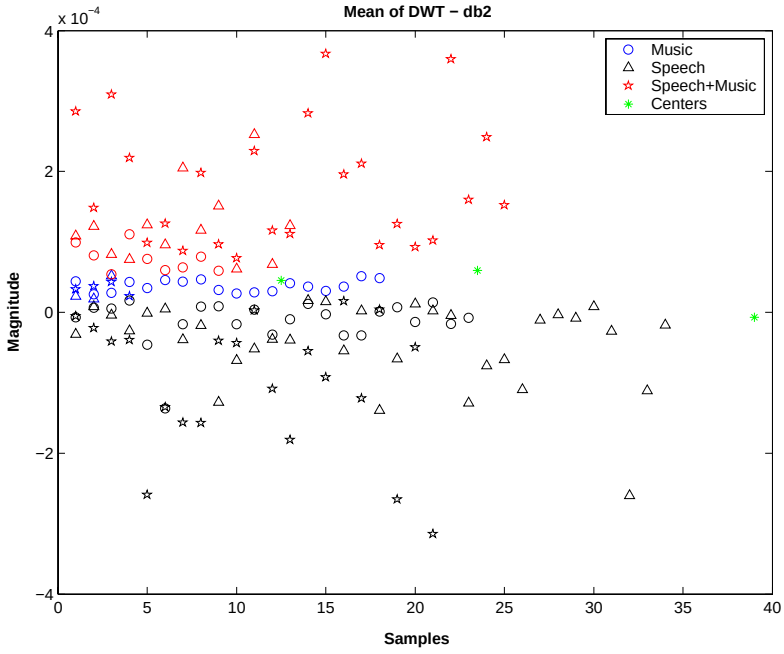

Figure 4.11: Clusters for Mean of DWT "DB2"

Table 4.9: Clustering results for Variance of DWT "DB2"

Variance of Discrete Wavelet Transform			
Cluster	***Classes***		
	Music	**Speech**	**Speech+Music**
Music	100%	52%	0%
Speech	0%	46%	34%
Speech+Music	0%	2%	66%
Total	100%	100%	100%

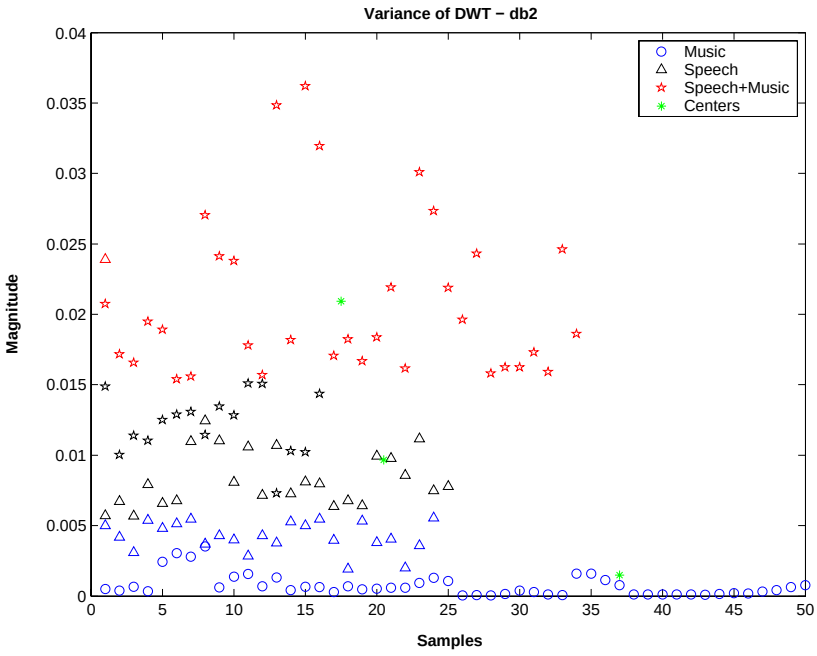

Figure 4.12: Clusters for Variance of DWT "DB2"

Table 4.10: Clustering results for %LEF

Percentage of Low Energy Frames			
Cluster	***Classes***		
	Music	**Speech**	**Speech+Music**
Music	66%	0%	4%
Speech	4%	96%	52%
Speech+Music	30%	4%	44%
Total	100%	100%	100%

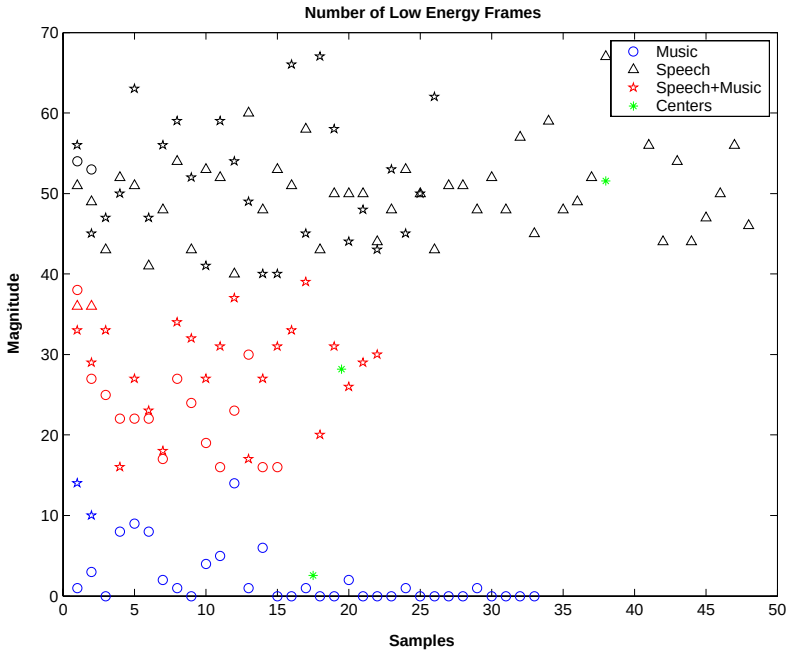

Figure 4.13: Clusters for %LEF

Table 4.11: Clustering results for R-ZC

Range of ZC			
Cluster	***Classes***		
	Music	**Speech**	**Speech+Music**
Music	100%	0%	6%
Speech	0%	90%	62%
Speech+Music	0%	10%	32%
Total	100%	100%	100%

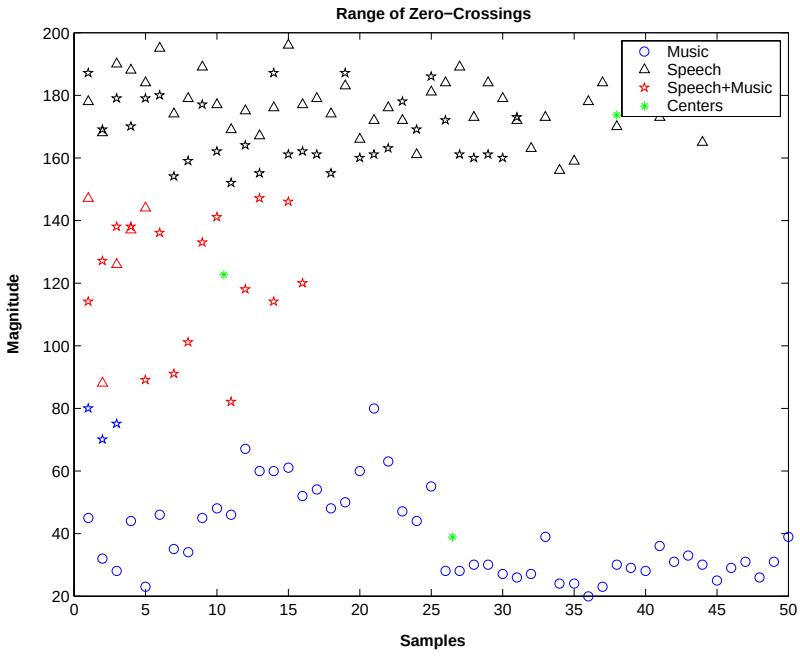

Figure 4.14: Clusters for R-ZC

Table 4.12: Clustering results for LPC

Linear Predictor Coefficients			
Cluster	***Classes***		
	Music	**Speech**	**Speech+Music**
Music	68%	14%	16%
Speech	0%	32%	20%
Speech+Music	32%	54%	64%
Total	100%	100%	100%

4.3 Contribution of Sets of Features to Classification

After studying the individual contribution of each feature in the classification process, we focus on choosing a subset of those features that maximize classification accuracy and at the same time reduce computational time. To achieve this, two methods can be used: *decision fusion* and *feature fusion*. Decision fusion based on a certain technique like majority voting uses individual features to classify, and then combine their classification result. Feature fusion puts different kinds of features in one feature space and makes a single decision. Selection of a proper feature subset is not an easy task. For this reason, we applied Fuzzy C-Mean clustering algorithm

on all possible sets of features. As it is not feasible to show the results for all the combinations of features, we have included results for sets of features that have given the highest clustering accuracy. After applying Fuzzy C-Means clustering algorithm we discovered that two different sets of features give the highest clustering accuracy. Both of them consist of three features among which two features are common, viz R-ZC and V-12MFCC. The third feature in the first set is SF and in the second set is %LEF. Per class accuracy for both sets was same as shown in Tables 4.13 and 4.14. After that, we repeated the same procedure again but this time we took V-4MFCC instead of V-12MFCC. In this case the highest clustering accuracy with minimum number of features was achieved with only one feature i.e. V-4MFCC as shown in Table 4.4.

Table 4.13: Clustering result for SF, R-ZC, and V-12MFCC

Set of three features			
Cluster	***Classes***		
	Music	**Speech**	**Speech+Music**
Music	98%	0%	4%
Speech	0%	86%	2%
Speech+Music	2%	14%	94%
Total	100%	100%	100%

Table 4.14: Clustering result for %LEF, R-ZC, and V-12MFCC

Set of three features			
Cluster	***Classes***		
	Music	**Speech**	**Speech+Music**
Music	98%	0%	4%
Speech	0%	86%	2%
Speech+Music	2%	14%	94%
Total	100%	100%	100%

After applying the clustering technique and short listing the potential discriminative features, we then apply a classification scheme on those features. In the next chapter we will discuss the classification frameworks that we have used in our research. We will also report the results achieved after applying those classification frameworks on different sets of short listed features.

Chapter 5

Classification Frameworks

After determining the features that will be used for classification, we need to specify a framework in which the classification process is to be carried out with. In this thesis, we have investigated two major approaches: Artificial Neural Networks and Hidden Markov Models. This chapter introduces both classification frameworks. Then, the experimentation using both approaches is detailed.

5.1 Artificial Neural Networks

Artificial Neural Networks (ANNs) have seen an explosion of interest over the last few years, and are being successfully applied on a range of problem domains. Indeed, wherever there are problems of prediction, classification or control, neural networks are being introduced. An ANN is a computational system inspired by the structure,

processing method and learning ability of a biological brain. ANNs are more robust at data analysis than statistical methods because of their ability to handle small variations of parameters and noise.

The concept of a neuron lies in the heart of ANN. A neuron, k, consists of three basic components, shown in Figure 5.1:

- A set of *synapses* or *connecting links*, each of which is assigned a weight of its own.
- An *adder* for summing the input signals, weighted by the respective synapses of the neuron. The j^{th} term in this summation is equal to the value of signal x_j at the input of synapse j multiplied by the synaptic weight w_{kj}
- An *activation function* for limiting the amplitude of the output of the neuron.

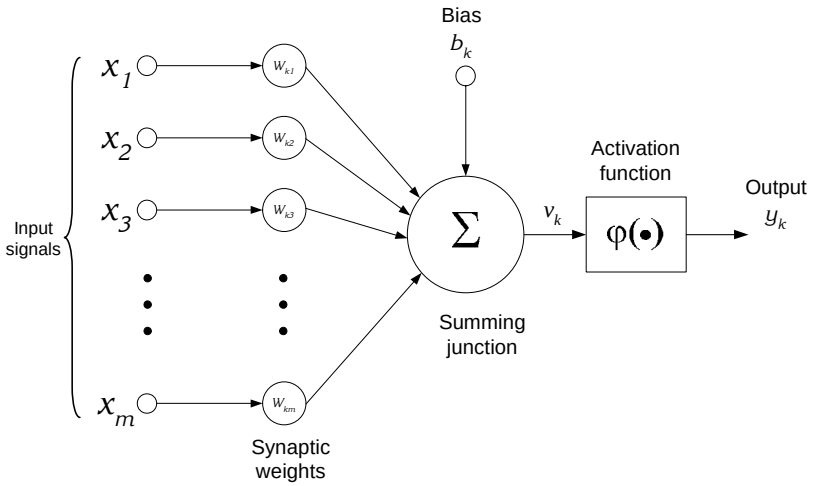

Figure 5.1: Nonlinear model of a neuron.

In mathematical terms, we may specify a neuron k by the following equations:

$$u_k = \sum_{j=1}^{m} w_{kj}\, x_j \tag{5.1}$$

$$v_k = u_k + b_k \tag{5.2}$$

$$y_k = \varphi(v_k) \tag{5.3}$$

where $x_1, x_2, \ldots, x_m$ are the input signals; $w_{k1}, w_{k2}, \ldots, w_{km}$ are the synaptic weights of neuron k; u_k is the *linear combiner output* due to the input signals;b_k is the bias which has the effect of applying an *affine transformation* to the output u_k in Equation 5.2, φ is the activation function; and y_k is the output signal of the neuron.

5.1.1 General Characteristics of ANNs

There are many types of ANNs. Listing all of them is outside the scope of this thesis. However, we can elaborate on important considerations that need to be taken into account when modeling an ANN. Generally the type of ANN depends on the following factors:

The first important characteristic we discuss is that of the *type of the learning algorithm used* in the ANN. With this regard, ANNs can be classified into those with supervised learning and those with unsupervised learning. In **supervised** learning, the actual output of a neural network is compared to the desired output. Weights, which are usually randomly set to begin with, are then adjusted by the network so

that the next iteration, or cycle, will produce a closer match between the desired and the actual output. The learning method tries to minimize the current errors of all processing elements. This global error reduction is created over time by continuously modifying the input weights until an acceptable network accuracy is reached. In **unsupervised** learning, ANNs use no external influences to adjust their weights. Instead, they internally monitor their performance. These networks look for regularities or trends in the input signals, and make adaptations according to the function of the network. Even without being told whether it's right or wrong, the network still must have some information about how to organize itself.

Another important characteristic of ANNs is that of *network topology.* ANNs can be classified, based on network topology into feedforward networks and feedback networks. In a **feedforward** ANN, signals travel in one direction only; from input to output as shown in Figure 5.2. There is no feedback, in the form of loops, in such networks. Therefore, the output of any layer does not affect that same layer. Feed-forward ANNs tend to be straight forward networks that associate inputs with outputs. They are extensively used in pattern recognition.

In a **feedback** ANN, signals travel in both directions by introducing loops in the network as shown in Figure 5.3. Feedback networks, also referred to as interactive or recurrent, are very powerful and can get extremely complicated.

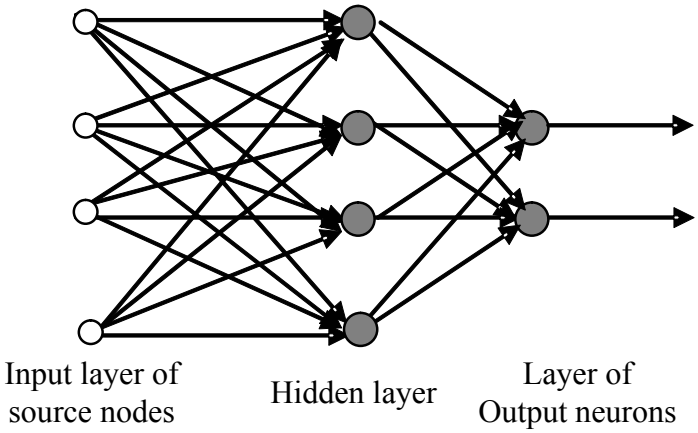

Figure 5.2: Fully connected feedforward network with one hidden layer and one output layer.

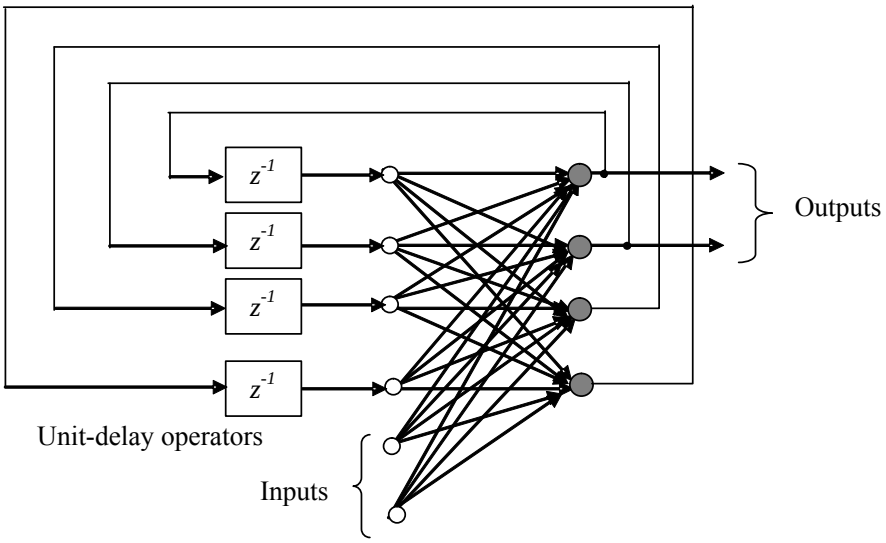

Figure 5.3: Recurrent network with hidden neurons.

In our work, we have only considered feedforward neural network topologies, namely the Multilayer Perceptron (MLP) and Radial Basis Functions (RBF). Next, we give an overview of MLP network, followed by an introduction to RBF network.

5.1.2 Multilayer Perceptron (MLP)

The Multilayer Perceptron (MLP) network is probably the most often considered member of the ANN family in classification. The main reason for this is its ability to model simple as well as very complex functional relationships. An MLP network consists of an input layer of source nodes, one or more hidden layers of computation nodes, and an output layer of computation nodes as shown in Figure 5.4. The input signal propagates through the network in a forward direction, on a layer-by-layer basis, thus considered a feedforward ANN.

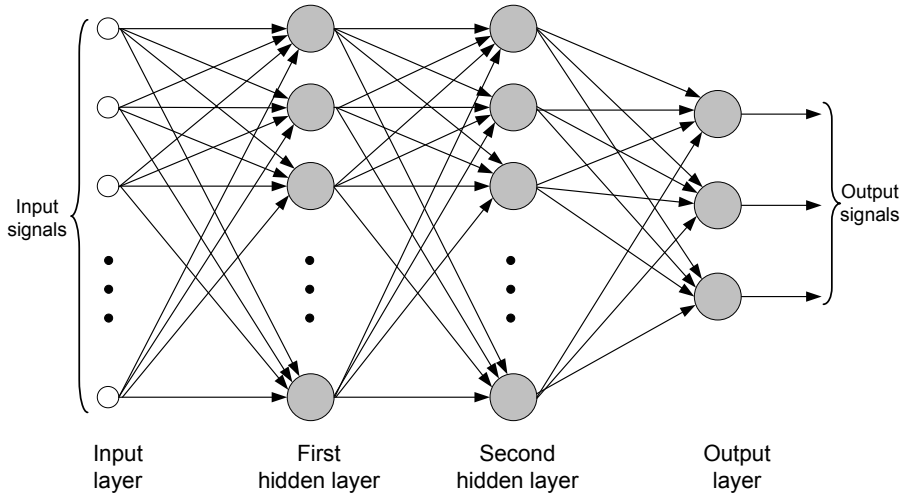

Figure 5.4: Multilayer perceptron with two hidden layers.

MLP networks successfully solve some difficult problems by training them in a *supervised* manner with a highly popular algorithm known as the *error back-propagation algorithm* or simply *back-propagation algorithm*.

Back-Propagation Algorithm

The back-propagation algorithm is based on the error-correction learning rule which requires preexisting training patterns, and involves a forward propagation step followed by a backward propagation step. In the *forward propagation* step, the input vector is fed into the nodes of the input layer and is propagated through the network, one layer at a time. A nonlinear activation function is used at each node for the transformation of the incoming signals to an output signal. This process repeats until the signals reach the output layer and an output vector is calculated. In the forward propagation step, the weights connecting the network nodes are fixed. During the *backward propagation* step, however, the synaptic weights are all adjusted based on an error-correction rule in which an *error signal*, $e_j(n)$ is produced, at the output neuron j for the n^{th} sample by simply subtracting the actual response of the network from the desired response. This error signal in turn propagates backward through the network so that the synaptic weights are adjusted in such a manner that the actual response of the network move closer to the desired response in a statistical sense. The mean-square error is usually used as a measure of the error which can be defined as:

$$\xi(n) = \sum_{j \in C} e_j^2(n) \tag{5.4}$$

where the set C consists of all the neurons in the output layer of the network, and the error $e_j(n)$ is calculated using the formula

$$e_j(n) = d_j(n) - f_j(n) \tag{5.5}$$

where $d_j(n)$ refers to the desired response from neuron j for the n^{th} sample and $f_j(n)$ is the generated response from neuron j for the n^{th} sample.

As the change in weights is evaluated by the delta rule:

$$\Delta w_{ji}(n) = \mu\ \delta_j(n)\ f_i(n) \tag{5.6}$$

where μ is the *learning-rate parameter* of the back-propagation algorithm and $\delta_j(n)$, the *local gradient* of neuron j for the n^{th} sample, is given by

$$\delta_j(n) = \begin{cases} e_j(n)\varphi\,'_j(v_j(n)) & \text{neuron } j \text{ is in the output layer} \\ \varphi\,'_j(v_j(n))\sum_k \delta_k(n) w_{kj}(n) & \text{neuron } j \text{ is in the hidden layer} \end{cases}$$

where $v_j(n)$ is given in Equation 5.2. Weights are updated as follows:

$$\underbrace{w_{ji}(n+1)}_{\text{New weights}} = \underbrace{w_{ji}(n)}_{\text{Old weights}} + \underbrace{\Delta w_{ji}(n)}_{\text{Change in weights}} \tag{5.7}$$

The performance of the back-propagation algorithm and MLP is governed by numerous factors involved in the design.

1. *Maximizing information content.* Every training sample presented to the back-propagation algorithm should be chosen on the basis that its information content is the largest possible for the task at hand [47].

2. *Choice of activation function.* The activation function $\varphi(v)$, defines the output of a neuron. The activation function usually used in MLP is a *sigmoidal nonlinearity*; which comes in two basic forms: logistic function and a hyperbolic tangent function [48].

   (a) *Logistic function.* The general form of this sigmoidal nonlinearity is defined by

   $$\varphi_j(v_j(n)) = \frac{1}{1+\exp(-av_j(n))}, \qquad a > 0 \text{ and } -\infty < v_j(n) < \infty \tag{5.8}$$

   The amplitude of the output in this nonlinearity lies in the range $0 \leq y_j \leq 1$.

   (b) *Hyperbolic tangent function.* This is another commonly used type of sigmoidal nonlinearity which in its most general form is defined by

   $$\varphi_j(v_j(n)) = a\tanh(bv_j(n)), \qquad (a,b) > 0 \tag{5.9}$$

   where a and b are constants. The hyperbolic tangent function is just the logistic function rescaled and biased. The amplitude of the output in this nonlinearity lies inside the range $-1 \leq y_j \leq +1$.

3. *Normalizing the inputs.* Each input variable should be *preprocessed* so that its mean value, averaged over the entire training set, is either close to zero, or small enough compared to its standard deviation [47].

4. *Initialization.* A good choice for the initial values of the synaptic weights of the network has a great impact on a successful network design. When the synaptic weights are assigned large initial values, it is highly likely that the neurons in the network will be driven into saturation, producing a constant activation, which will cause the training of the network to become stuck near the starting point. However, if the synaptic weights are assigned very small initial values, the back-propagation algorithm may operate on a very flat area around the origin of the error surface. The back-propagation algorithm uses gradient descent to find the global minimum. Due to the flat area around the origin of the error surface the performance improvement of the classification will drop to zero and hence the learning process terminates. For these reasons the use of both large and very small values for initializing the synaptic weights should be avoided [48].

5. *Setting the learning rates.* All neurons in the MLP should ideally learn at the same rate. According to [47], it is suggested that for a given neuron, the learning rate μ should be inversely proportional to the square root of synaptic connections made to that neuron. If k is the number of neurons in the previous

layer then μ is given by

$$\mu \propto \frac{1}{\sqrt{k}}$$

6. *Setting the number of hidden neurons.* Deciding the number of hidden neurons is considerably difficult. According to [49], the number of hidden neurons should never exceed twice the number of input layer units, but this number may not be known in advance. The number of hidden neurons controls the degree of generalization in the network. As the number of hidden neurons is increased, the accuracy of input recognition increases, but the capacity for generalization decreases. When the number of hidden units approaches the number of samples, the network can recognize every different sample exactly, but has no ability to generalize. In other words a large number of hidden neurons can lead to overfitting of the training data.

7. *Setting the number of hidden layers.* The number of hidden layers required depends on the complexity of the relationship between the inputs and the outputs. If the input/output relationship is linear (can be approximated by a straight line graph), the network does not need a hidden layer at all. It is unlikely that any practical problem will require more than two hidden layers [49]. In theory, an MLP with one hidden layer is sufficient to approximate any continuous function [50].

5.1.3 Radial Basis Functions (RBF)

RBFs are feedforward network that are used in a wide variety of contexts such as function approximation, pattern recognition and time series prediction. In these networks the learning involves only one layer with lesser computations. This results in reduction in the training time in contrast to MLP that uses back propagation algorithm to update the weights of all the layers. These features make RBF attractive in many practical problems.

The construction of an RBF network, in its most basic form, consists of three layers: the input layer of source nodes, middle layer which is the only hidden layer in the network that applies a nonlinear transformation and the output layer which is linear as shown in Figure 5.5. Every input node is connected to all the nodes in the hidden layer through unity weights (direct connection). Each of the hidden layer nodes is connected to the output node through some weights $\lambda_{11}, \lambda_{12}, \ldots, \lambda_{1n_o}$, where n_o is the number of neurons in the output layer, and λ_{ij} represents the weight of the connection between the i^{th} neuron in the hidden layer and the j^{th} neuron in the output layer. Each hidden neuron finds the distance, normally using the Euclidean distance, between the input, X, and its center, y, and passes the resulting scalar through a non-linearity $\varphi()$. So the output of the hidden neuron is given by $\varphi(\|X_n - y_i\|)$, where X_n is the n^{th} sample where n ranges from 1 to m, the total

number of samples, y_i is the center of the i^{th} hidden neuron where $i = 1, 2, \ldots, n_h$, and $\varphi(\cdot)$ is the nonlinear basis function. Normally this function is taken as a Gaussian function of width β. The center of the hidden neuron must be chosen carefully as the classification accuracy of RBF network depends on the choice of centers.

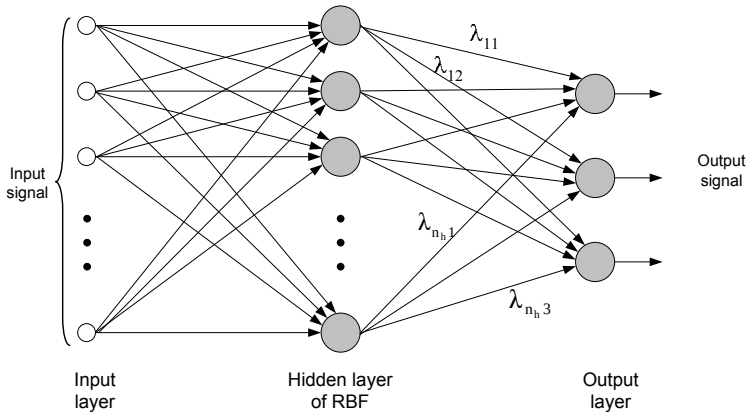

Figure 5.5: A general RBF network.

Learning Algorithm

The objective of the learning algorithm is to minimize the mean-square error, given by

$$\xi(n) = \sum_{j=1}^{n_o} e_j^2(n) \tag{5.10}$$

where $e_j(n)$ is the classification error at the output of neuron j for n^{th} sample. As with the case of MLP network, the classification error at the output layer is given by

$$e(n) = d_n - f_n, \qquad (\text{for } n^{th} \text{ sample}) \tag{5.11}$$

where d_n is the desired output and f_n is the generated output for the n^{th} sample.

Let m denote the total number of samples, n_i denote the number of neurons in the input layer, n_h denote the number of neurons in the hidden layer, and n_o denote the number of neurons in the output layer. Then, the value of f_n is given by

$$f_n = \lambda \, \mathbf{A_n}, \qquad n = 1, 2, \cdots, m \tag{5.12}$$

where λ is set of weights between hidden layer and output layer,

$$\lambda = \begin{bmatrix} \lambda_{11} & \lambda_{21} & \cdots & \lambda_{n_h 1} \\ \lambda_{12} & \lambda_{22} & \cdots & \lambda_{n_h 2} \\ \vdots & \vdots & & \vdots \\ \lambda_{1 n_o} & \lambda_{2 n_o} & \cdots & \lambda_{n_h n_o} \end{bmatrix}$$

and A_n is the *influence matrix.*

$$\mathbf{A_n} = \begin{bmatrix} a_{n1} \\ a_{n2} \\ \vdots \\ a_{n n_h} \end{bmatrix}$$

Each element of the influence matrix can be defined as follows

$$a_{n1} = \varphi(\|X_n - y_1\|)$$

$$a_{n2} = \varphi(\|X_n - y_2\|)$$

$$\vdots$$

$$a_{n n_h} = \varphi(\|X_n - y_{n_h}\|)$$

where, $\varphi(\|X_n - y_i\|)$ is calculated as $\exp(-\frac{\|X_n - y_i\|^2}{\beta^2})$, $n = 1, 2, \cdots, m$, $i = 1, 2, \cdots, n_h$, and β is the width of the Gaussian function. So in general, we can write Equation 5.12 as

$$f = \lambda \, \mathbf{A} \tag{5.13}$$

where,

$$f = \begin{bmatrix} f_1 & f_2 & \cdots & f_m \end{bmatrix}$$

and

$$A = \begin{bmatrix} A_1 & A_2 & \cdots & A_m \end{bmatrix}$$

Since the desired response d is given by

$$d = \lambda \, \mathbf{A}$$

the value of λ can be computed from

$$\lambda = (\mathbf{A^T} \, \mathbf{A})^{-1} \mathbf{A^T d} \tag{5.14}$$

where $(A^T \; A)^{-1} A^T$ is called the pseudo inverse of matrix A.

5.1.4 Comparison of MLP and RBF networks

Both MLP and RBF are nonlinear layered feedforward networks. In addition both of them are universal approximators. These two networks differ from each other in many aspects, some of them are [48]:

1. In its basic form an RBF network has a single hidden layer, whereas an MLP may have one or more hidden layers.

2. The hidden layer of an RBF network is nonlinear, whereas the output layer is linear. In MLP the hidden and the output layers are usually nonlinear.

3. MLP constructs *global* approximations to nonlinear input/output mapping, whereas the RBF constructs *local* approximations to nonlinear input/output mapping.

4. In MLP the neurons located in a hidden layer or an output layer usually share a common neuronal model. However, neurons in the hidden layer of an RBF network are quite different and serve a different purpose from those in the output layer of the network.

5. The argument of the activation function of each hidden neuron in an RBF network computes the *Euclidean norm (distance)* between the input vector and the center of that neuron. Whereas, the activation function of each hidden neuron in an MLP computes the *inner product* of the input vector and the synaptic weight vector of that neuron.

5.2 Hidden Markov Models

Hidden Markov Models belong to a class of statistical models that employ the statistical properties of the signal in carrying out recognition and/or classification. Other statistical models in this domain include Gaussian processes, Poisson processes, and Markov processes. In this section we will briefly discuss the theory of Hidden Markov Models (HMMs). A more detailed description of this topic can be found in [51, 52]. According to [51]:

> *An HMM is a doubly stochastic process with an underlying stochastic process that is not observable (it is hidden), but can only be observed through another set of stochastic processes that produce the sequence of observed symbols.*

A Markov chain or process is a sequence of events, usually called states, the probability of each of which is dependent only on events immediately preceding it. A Hidden Markov Model (HMM) represents stochastic sequences as Markov chains where the states are not directly observed, but are associated with a probability density function (pdf). The generation of a random sequence is then the result of a random walk in the chain and of a draw (called an emission) at each visit of a state. The sequence of states, which is the quantity of interest in most of the pattern recognition problems, can be observed only through the stochastic processes defined in each state. In other words, we must know the parameters of the pdfs of each state

before being able to associate a sequence of states $Q = \{q_1, ..., q_T\}$ to a sequence of observations $O = \{O_1, O_2, ..., O_T\}$, where T is the length of state sequence or the number of observations. The true sequence of states is therefore hidden by a first layer of stochastic processes.

An HMM λ is defined by its parameters $\lambda = (A, B, \pi)$. π stands for the vector of the initial state-transition probabilities, the NxN matrix A represents the state-transition probabilities a_{ij} from state s_i to state s_j and finally B denotes the vector of the output densities $b_i(x)$ (usually Gaussian or combination of Gaussians) of each state s_i. There are several important points of how to model the outputs of the experiment via HMMs. One of the most difficult parts of the modeling procedure is to decide on the size of the model i.e. the number of states. Without some a-priori information, this decision is often difficult to make and could involve trial and error before settling on the most appropriate model size. The other aspects of the modeling procedure consists of state transition probabilities, probabilities of each state, and the size of the observation sequence.

5.2.1 The Three Problems for HMMs

Given the form of the HMM discussed in the previous section, there are three key problems of interest that must be solved for the model to be useful in real world applications. These problems are the following:

Problem 1 (Evaluation Problem) [51]: *Given the observation sequence* $O = O_1, O_2, ..., O_T$, *and the model* $\lambda = (A, B, \pi)$, *how do we efficiently compute* $P(O|\lambda)$, *the probability of the observation sequence, given the model?* To solve this problem there exists an efficient procedure called forward-backward procedure which allows us to choose the model which best matches the observations.

Problem 2 (Decoding Problem) [51]: *Given the observation sequence* $O = O_1, O_2, ..., O_T$, *and the model* λ, *how do we choose a corresponding state sequence* $Q = q_1, q_2, ..., q_T$ *which is optimal in some meaningful sense?* This is the one in which we attempt to uncover the hidden part of the model, i.e. the state sequence. This is a typical estimation problem. We usually use an optimality criterion to solve this problem as best as possible. Unfortunately, there are several possible optimality criteria that can be imposed and hence the choice of criterion is a strong function of the intended use for the uncovered state sequence. A typical use of the recovered state sequence is to learn about the structure of the model, and to get average statistics, behavior, etc. within individual states. A formal technique for finding the single best state sequence exists and is called the Viterbi algorithm.

Problem 3 (Learning Problem) [51]: *How do we adjust the model parameters* $\lambda = (A, B, \pi)$ *to maximize* $P(O|\lambda)$*?* Here we attempt to optimize the model parameters so as to best describe how the observed sequence comes about. We call

this a training sequence in this case since it is used to train the model. The training problem is the crucial one for most applications of HMM's. Since it allows us to optimally adapt model parameters to observed training data i.e. to create best models for real phenomena. To solve this problem an iterative procedure such as Baum-Welch method, equivalently the Expectation-Maximization (EM) method, or gradient techniques is used.

5.2.2 Types of HMM

HMMs can be classified according to two different aspects:

1. ***The nature of the elements of the B matrix, which are distribution functions:*** Distributions are defined on finite spaces in the so called *discrete HMMs*. In this case, observations are vectors of symbols in a finite alphabet of N different elements. For each one of the vector components, a discrete density is defined, and the distribution is obtained by multiplying the probabilities of each component. Another possibility is to define distributions as probability densities on continuous observation spaces. The most popular approach is to characterize the model transitions with mixtures of base densities which are usually Gaussian, and can be parameterized by the mean vector and the covariance matrix. HMMs with these kinds of distributions are usually referred to as *continuous HMMs*.

2. ***State transitions:*** A general HMM is assumed to have a full state transition matrix, i.e. transitions can be made, from any state in some way to any other state. Such models are *ergodic* in the sense that any state will be revisited with probability one and that such revisits are not required to take place at periodic intervals of time as shown in Figure 5.7. For some applications we are interested in non-ergodic models in which transitions can only be made to a state whose index is as large or larger than the index of the current state. Such models have been called *left-to-right models* since the state sequence which produced the observation sequence must always proceed from the left-most state to the rightmost state as shown in Figure 5.6. Such left-to-right models inherently impose a temporal order to the HMM since lower numbered states account for observations occurring prior to those for higher numbered states.

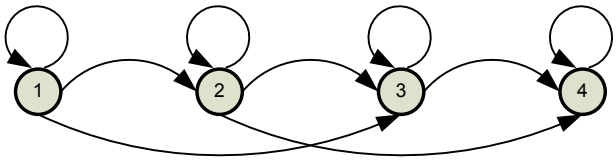

Figure 5.6: A left-to-right hidden Markov model

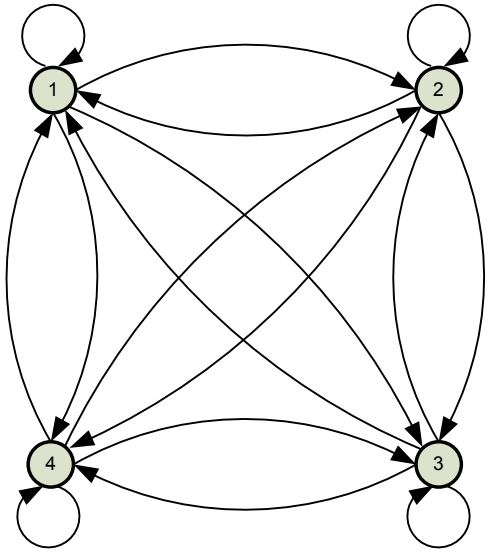

Figure 5.7: An ergodic hidden Markov model

5.3 Experimental Setup

In order to successfully carry out the classification process using either of the frameworks, discussed in the previous section, one should carefully choose a set of features that will be able to discriminate between the desired classes with highest accuracy. The data that needs to be classified in our case consists of audio signals. More specifically, we interested in classifying audio signals into three categories: speech, music, and speech/music.

In order for such a classifier to be successful, it has to possess certain characteristics. First, due to the variability of music and speech signals, the classifier must

be able to generalize from a relatively little amount of training data. Second, the notion of speech and music may differ from one application to another. For example, speech with background music may be considered as speech or music depending on the relative volume of the music as compared to speech. Hence, the classifier must adapt to different situations in order to give accurate results. In addition, since audio data is composed of a large amount of information size-wise, it requires any practical classifier to be fast and simple.

In this section, we present our experimental setup expending on the Audio Database used and the classification parameters chosen for each classification framework. In the next section, we present our findings on applying these classifiers on a set of features and conclude by listing the most successful set of features in terms of accuracy and compare and contrast the three classification frameworks used.

5.3.1 Experimental Apparatus

The work done here was implemented on Matlab®[1] 6.5 on an Intel® Pentium® 4 system with 512 MB RAM on Microsoft® Windows® XP[2] platform. Audio samples were extracted from movies, documentaries, and different speeches by using Video To Audio Converter 1.0 and Sony® Sound Forge®[3] 7.0. We have used 16-bit, 44.1

[1]Matlab® is a trademark of The MathWorks, Inc.
[2]Windows® and Windows XP are the trademarks of Microsoft Corporation.
[3]Sound Forge® is a trademark of Sony Media Software.

kHz, mono PCM wave files, as audio samples.

5.3.2 Database

The experiments described in this section were carried out using a database of music, speech, and speech+music. All of the speech and speech+music material was conversational and included examples from both genders. The following languages were represented: American English, Pakistani Urdu, Japanese, Spanish, and Hebrew. The audio samples were extracted from documentaries and from different movies. There were approximately 2.25 hours of speech, 2.72 hours of music and 0.62 hours of speech/music data distributed over 3sec audio files as shown in Table 5.1. All audio data were sampled at 44.1kHz rate and were 16-bit, mono PCM wave files.

Table 5.1: Audio data samples

Language	Number of samples		
	Music	Speech	Speech/Music
English	-	50	50
Urdu	-	1543	100
Japanese	-	427	336
Spanish	-	542	154
Hebrew	-	140	100
Total	3268	2702	740

5.3.3 MLP Classifier

The MLP classifier consists of three layers: an input layer, one hidden layer, and an output layer. The input layer contains a number of neurons equals to the number of features considered per audio sample. The reason behind our choice of one hidden layer is the fact that continuous feedforward neural networks with a single hidden layer and a nonlinear sigmoidal activation function provide good approximations to arbitrary decision regions [50, 53].

The output layer consists of three neurons, each corresponding to a class (Music, Speech, and Speech/Music). The MLP has been chosen to be *fully connected*, i.e., a neuron in any layer is connected to all neurons in the previous layer.

Prior to training, small random numbers have been generated to initialize weights on each communication link, called connection, between neurons. In addition, the input features have been normalized as neural networks risk saturation if feature vectors contain values higher than 1. Saturation refers to the situation where synaptic weights change slowly causing a very long training time. With regard to the number of neurons in the hidden layer, we have used 5 neurons and 10 neurons. After carrying out the classification process and comparing the results, we decided to work with 5 neurons only as the accuracy was not greatly affected by the increase in

number though the processing time of the classifier increased substantially. Due to nonlinear behavior of patterns, we have used sigmoidal functions as an activation function. We have used tan sigmoid function with output values between -1 and 1 for the hidden neurons, and log sigmoid function for the output neurons, with values ranging between 0 and 1.

5.3.4 RBF Classifier

A *reduced RBF* classifier has been considered in experimentation. A reduced RBF network is an RBF network in which the number of centers is less than the total number of input samples, as opposed to a *complete RBF*, where the number of centers is equal to that of the input samples. The number of centers used equals to 3, the number of classes. We have extracted the centers from the input features by using the Fuzzy C-Means clustering algorithm. As a learning algorithm we have used the average square error algorithm.

5.3.5 HMM Classifier

In order to compare results obtained from using neural networks to those using statistical models like HMMs, we have trained 3 HMMs, one per class. After computing the log-likelihood that a single sample generates if the i^{th} model where $1 \leq i \leq 3$ gives the highest value the sample is classified to belong to class i. This is called sequence classification.

It is worth mentioning that an important motive behind using HMMs is that many other researchers in this area have used statistical classifiers, including Gaussian Mixture Models (GMM). As there is no set of benchmark data that is available for comparing the performance of these classifiers, we thought that including a classifier from this category of classifiers can be considered the next to best attempt to hold such a comparison.

5.4 Experimental Results

First, we have examined 7 features:

1. *RMS of Lowpass Signal (RMS-LPS)*,
2. *Mean of Discrete Wavelet Transform (M-DWT)*,
3. *Variance of Discrete Wavelet Transform (V-DWT)*,
4. *Spectral Flux (SF)*,
5. *Percentage of Low Energy Frames (%LEF)*,
6. *Range of Zero Crossings (R-ZC)*, and
7. *Linear Predictive Coefficients (LPC)*

and applied all the three classifiers on these features. Table 5.2 and Figure 5.8 shows the accuracies achieved by applying MLP, RBF, and HMM.

Table 5.2: Classification results for RMS-LPS, M-DWT, V-DWT, SF, %LEF, R-ZC, and LPC

Language	Accuracy with MLP				Accuracy with RBF				Accuracy with HMM			
	M	S	S+M	Total	M	S	S+M	Total	M	S	S+M	Total
English	100	93.33	93.33	95.55	100	100	30	76.67	100	84	73.33	85.78
Urdu	73.33	76.66	53.33	67.77	90	25	5	40	82	97.34	44	74.45
Japanese	89	85	31	68.33	86.57	94.03	0	60.20	58.40	76.60	44.20	59.73
Spanish	93.48	63.04	26.09	60.87	70	43.33	3.33	38.89	99.13	57.39	56.09	70.87
Hebrew	11.54	80.77	7.7	33.34	95	70	15	60	96.67	69.33	54.67	73.56
All	85.85	82.65	32.42	66.97	77.70	68.92	0	48.87	91.17	71.71	52.61	71.83

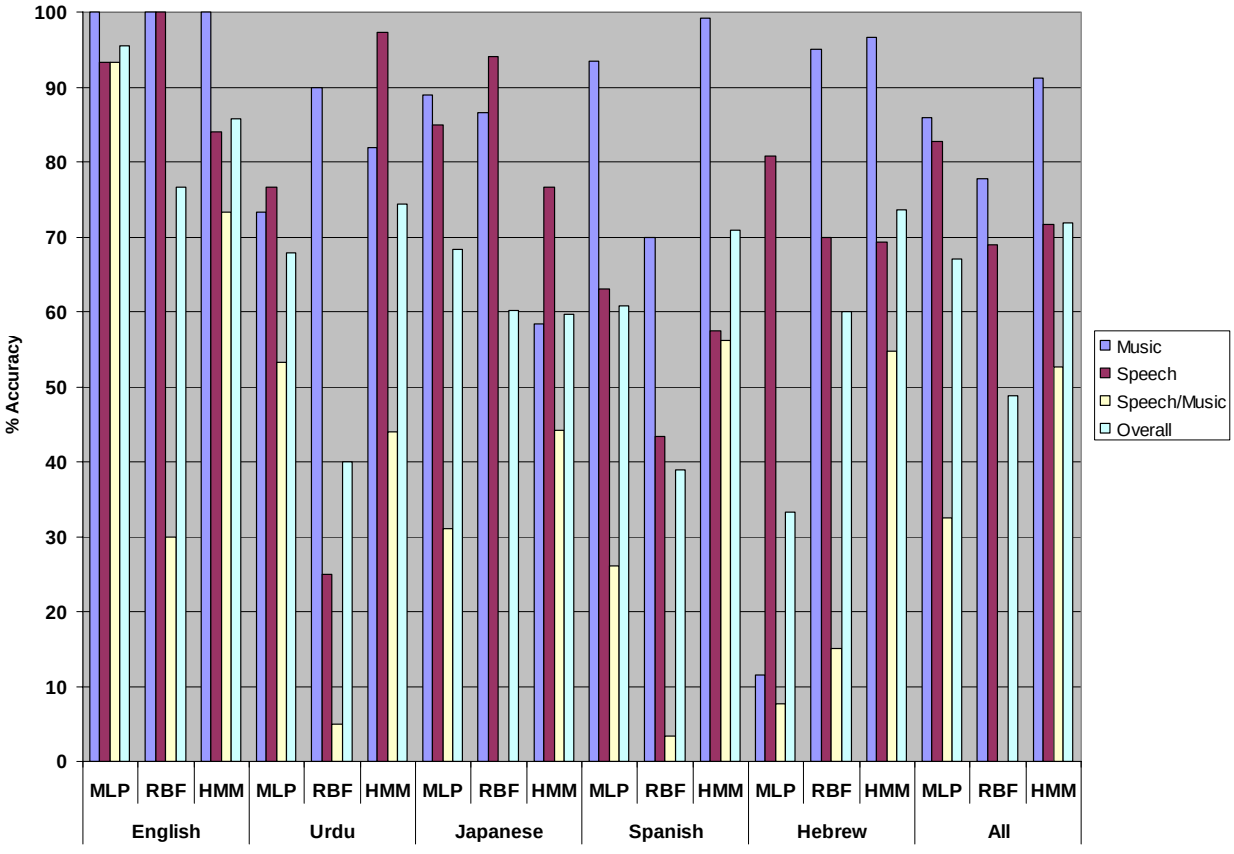

Figure 5.8: Accuracy with features: RMS-LPS, M-DWT, V-DWT, SF, %LEF, R-ZC, and LPC

Table 5.3: Classification results for RMS-LPS, V-DWT, SF, R-ZC, and LPC

Language	Accuracy with MLP				Accuracy with RBF				Accuracy with HMM			
	M	S	S+M	Total	M	S	S+M	Total	M	S	S+M	Total
English	100	100	70	90	100	100	70	90	100	58.67	93.33	84
Urdu	70	95	25	63.33	55	75	30	53.33	34.67	96.67	52.67	61.33
Japanese	91.04	82.09	32.83	68.65	85.07	89.55	0	58.21	82.80	70.60	33.20	62.20
Spanish	93.33	0	26.67	40	76.67	36.67	0	37.78	94.35	33.48	72.61	66.81
Hebrew	100	75	0	58.33	95	70	0	55	93.33	51.33	58.67	67.78
All	88.51	27.70	62.84	59.68	33.78	33.78	16.90	28.15	87.48	57.21	49.48	64.72

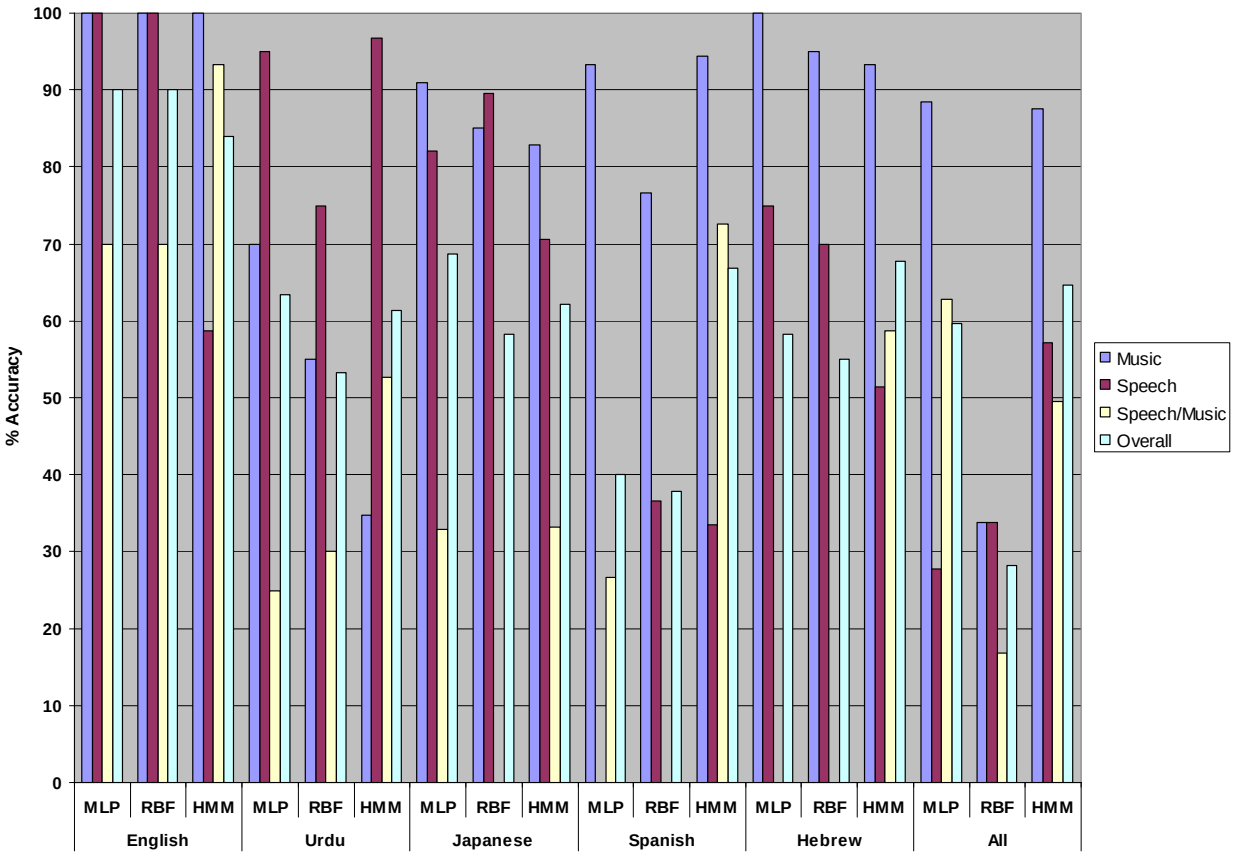

Figure 5.9: Accuracy with features: RMS-LPS, V-DWT, SF, R-ZC, and LPC

As we discussed in the previous chapter, when we applied Fuzzy C-Mean clustering algorithm on each of the features we found that M-DWT, and %LEF are not performing well, so we removed M-DWT and %LEF and then applied the classification process again and received the results given in Table 5.3 and Figure 5.9.

While we were examining the aforementioned features we added another feature which is variance of MFCC. Typically 12 coefficients are used for speech representation that is why we have taken 12 coefficients of MFCC (V-12MFCC) for this task. Now, we tried to find out the classification accuracies for all the classifiers when this feature was used with all 7 features previously used and also with 5 features i.e. after removing M-DWT and %LEF. The results achieved are produced below in Table 5.4 and Figure 5.10 when V-12MFCC was used with 7 features and in Table 5.5 and Figure 5.11 when V-12MFCC was used with 5 features.

When we were investigating V-12MFCC we found that the clustering result achieved by using only first 4 coefficients of MFCC (V-4MFCC) was the same as that for 12 coefficients (see Section 3.2.2 in Chapter 3 on page 41). Therefore, to find out the classification accuracies for all the classifiers when V-4MFCC is used with all 7 features previously used and also with 5 features i.e. after removing M-DWT and %LEF, we repeated the same procedure followed for V-12MFCC again. The results achieved are shown below in Table 5.6 and Figure 5.12 when V-4MFCC was used

with 7 features and in Table 5.7 and Figure 5.13 when V-4MFCC was used with 5 features.

In Chapter 3 we discussed that we applied Fuzzy C-Mean clustering algorithm on all possible combinations of features to find out the best combination which gives the highest accuracy for classification. It came to the fact that there are two different sets each containing three features that might give the highest classification accuracy. The first set of features was SF, R-ZC, and V-12MFCC, and the second set of features contains %LEF, R-ZC, and V-12MFCC. As we have made an assertion before that some features work better when used with others that is why it seems that the rest of the coefficients of V-12MFCC - 5th coefficient and onwards - are after all not useless and may contribute in the process of classification when used with other features.

We applied MLP, RBF, and HMM on both of these sets of features to find the classification accuracy. Table 5.8 and Figure 5.14 represents the results of classification when the features: SF, R-ZC, and V-12MFCC were used. Table 5.9 and Figure 5.15 represents the results of classification when the features: %LEF, R-ZC, and V-12MFCC were used.

Table 5.4: Classification results for RMS-LPS, M-DWT, V-DWT, SF, %LEF, R-ZC, LPC, and V-12MFCC

Language	Accuracy with MLP				Accuracy with RBF				Accuracy with HMM			
	M	S	S+M	Total	M	S	S+M	Total	M	S	S+M	Total
English	100	100	90	96.67	100	100	90	96.67	100	100	90.67	96.89
Urdu	95	90	100	95	25	65	0	30	55.33	100	64	73.11
Japanese	92.54	89.55	70.15	84.08	2.98	10.45	7.46	6.96	97	88	38.20	74.40
Spanish	93.33	86.67	23.33	67.78	80	30	23.33	44.44	95.65	34.78	78.26	69.56
Hebrew	95	65	75	78.33	90	0	25	38.33	91.33	70	28	63.11
All	86.48	57.43	60.81	68.24	76.35	31.76	0	36.04	97.66	52.79	80.81	77.09

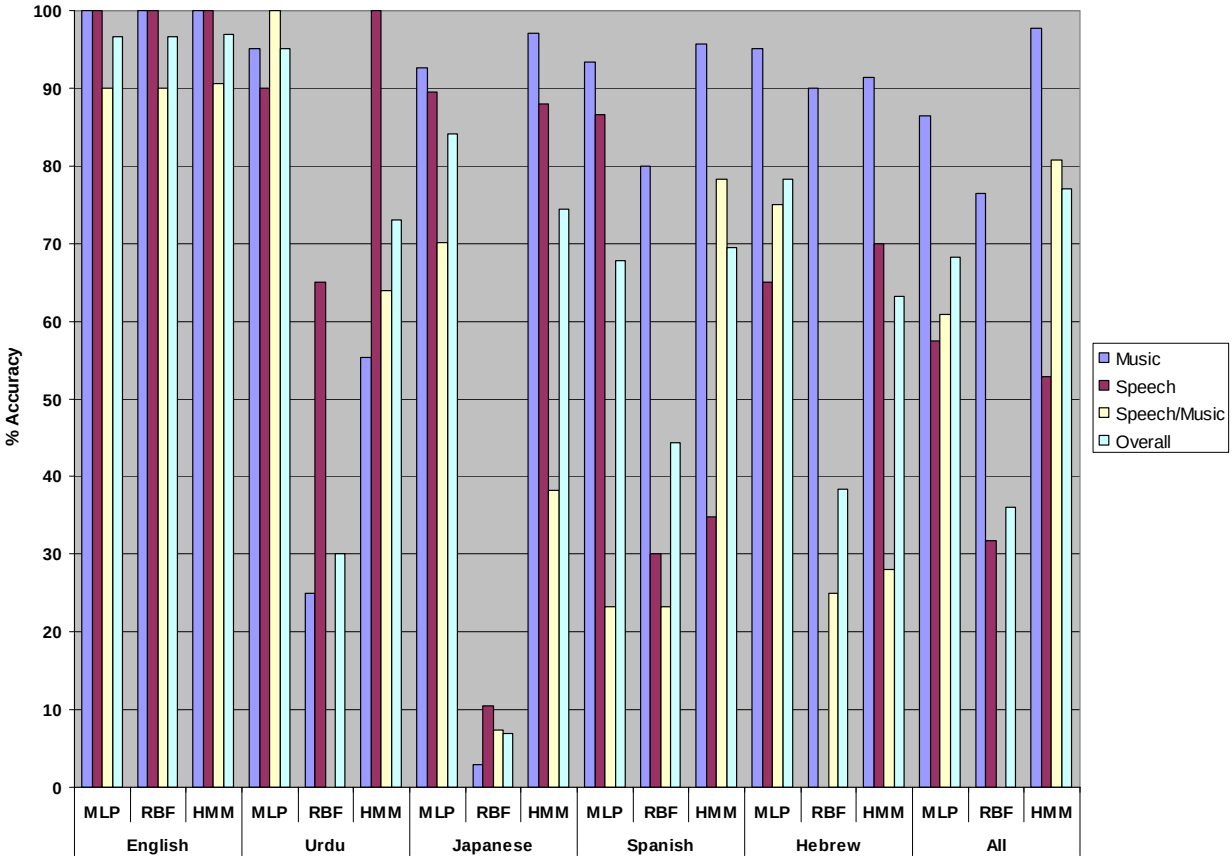

Figure 5.10: Accuracy with features: RMS-LPS, M-DWT, V-DWT, SF, %LEF, R-ZC, LPC, and V-12MFCC

Table 5.5: Classification results for RMS-LPS, V-DWT, SF, R-ZC, LPC and V12-MFCC

Language	Accuracy with MLP				Accuracy with RBF				Accuracy with HMM			
	M	S	S+M	Total	M	S	S+M	Total	M	S	S+M	Total
English	100	100	90	96.67	100	100	90	96.67	100	81.33	81.33	87.56
Urdu	80	100	95	91.67	25	60	0	28.33	47.33	98	42.67	62.67
Japanese	100	89.55	61.20	83.58	0	4.48	10.45	4.98	87.20	76	34.60	65.93
Spanish	93.33	83.33	43.34	73.33	80	26.67	23.33	43.33	94.78	47.83	67.39	70
Hebrew	95	70	75	80	90	0	45	45	66	57.33	42.67	55.33
All	86.48	37.16	78.38	67.34	71.62	23.65	0	31.76	91.62	49.01	62.25	67.63

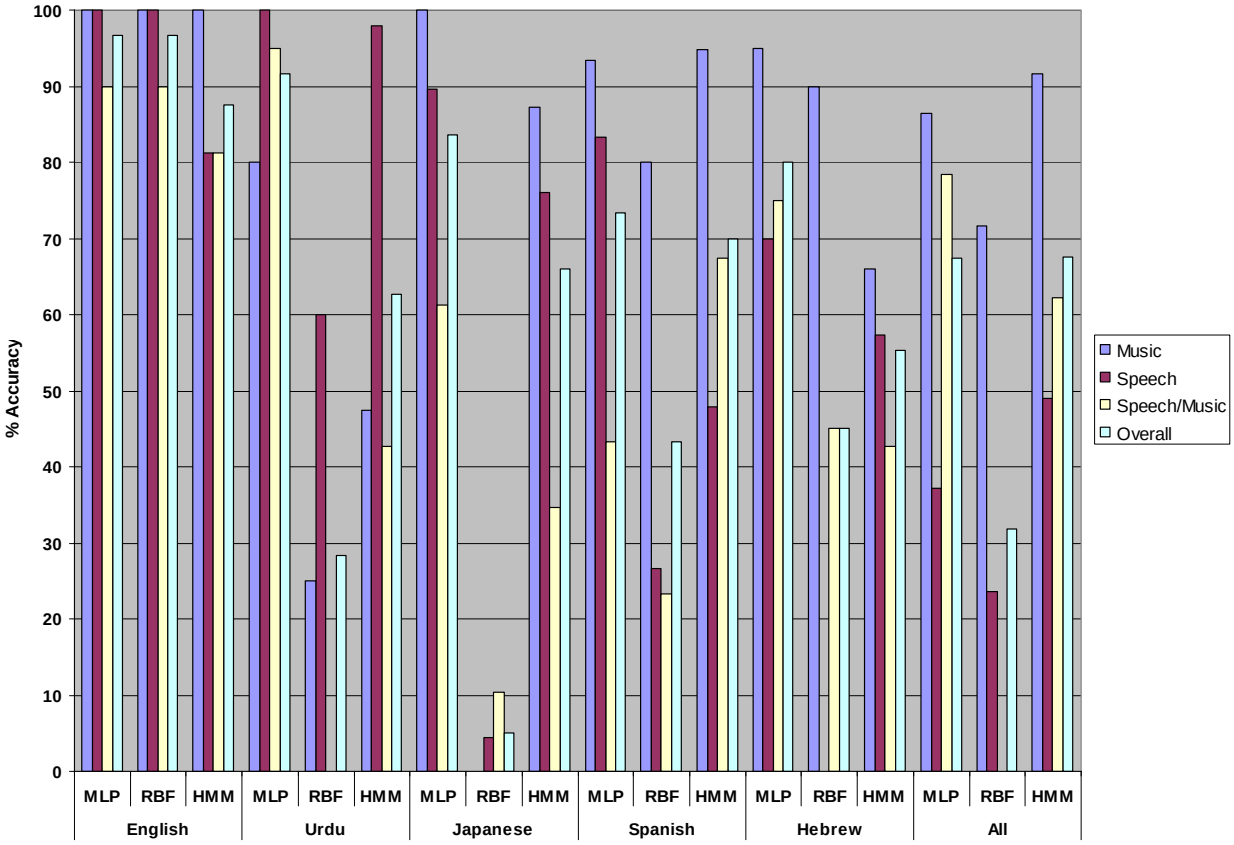

Figure 5.11: Accuracy with features: RMS-LPS, V-DWT, SF, R-ZC, LPC and V12-MFCC

Table 5.6: Classification results for RMS-LPS, M-DWT, V-DWT, SF, %LEF, R-ZC, LPC, and V-4MFCC

Language	Accuracy with MLP				Accuracy with RBF				Accuracy with HMM			
	M	S	S+M	Total	M	S	S+M	Total	M	S	S+M	Total
English	100	100	93.33	97.78	100	100	90	96.67	100	66.67	84	83.56
Urdu	90	65	85	80	15	50	25	30	68.67	96.67	49.33	71.56
Japanese	82.1	95.52	71.64	83.09	2.98	5.97	11.94	6.96	73	84	31.20	62.73
Spanish	86.67	90	16.67	64.45	76.67	30	26.67	44.45	90.44	44.35	62.61	65.80
Hebrew	95	60	90	81.67	90	0	20	36.67	89.33	84.67	27.33	67.11
All	86.04	75.67	39.19	66.97	72.97	28.38	0	33.78	94.32	84.05	33.51	70.63

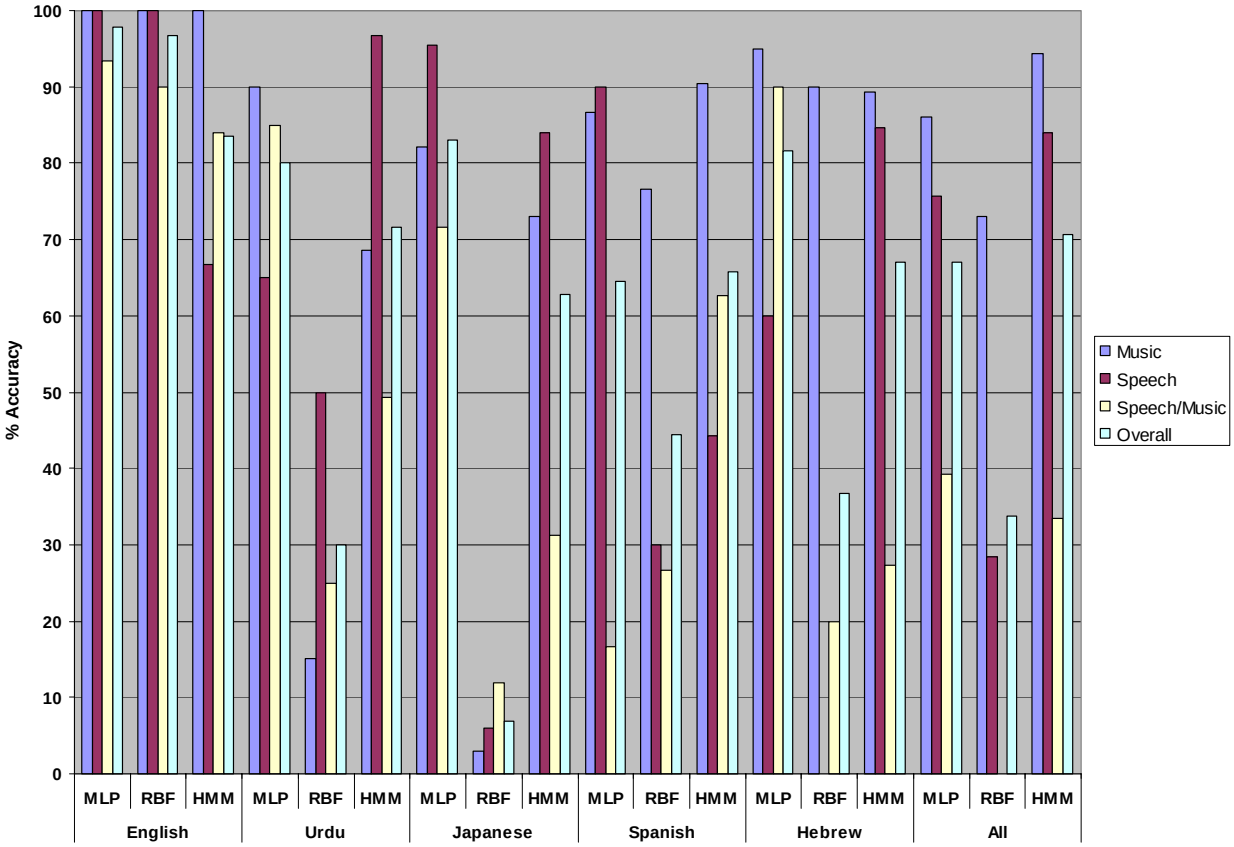

Figure 5.12: Accuracy with features: RMS-LPS, M-DWT, V-DWT, SF, %LEF, R-ZC, LPC, and V-4MFCC

Table 5.7: Classification results for RMS-LPS, V-DWT, SF, R-ZC, LPC, and V-4MFCC

Language	Accuracy with MLP				Accuracy with RBF				Accuracy with HMM			
	M	S	S+M	Total	M	S	S+M	Total	M	S	S+M	Total
English	100	100	100	100	100	100	90	96.67	100	89.33	85.33	91.56
Urdu	100	95	40	78.33	0	45	25	23.33	60.67	96.67	42	66.64
Japanese	95.52	85.07	71.64	84.08	0	2.98	11.94	4.97	94.80	79.80	25.60	66.73
Spanish	93.33	43.33	56.67	64.44	80	26.67	30	45.56	97.39	50	66.52	71.31
Hebrew	95	80	65	80	90	0	45	45	48	76.66	19.33	48
All	83.11	39.86	64.2	62.39	70.27	20.27	0	30.18	95.32	95.32	7.30	65.98

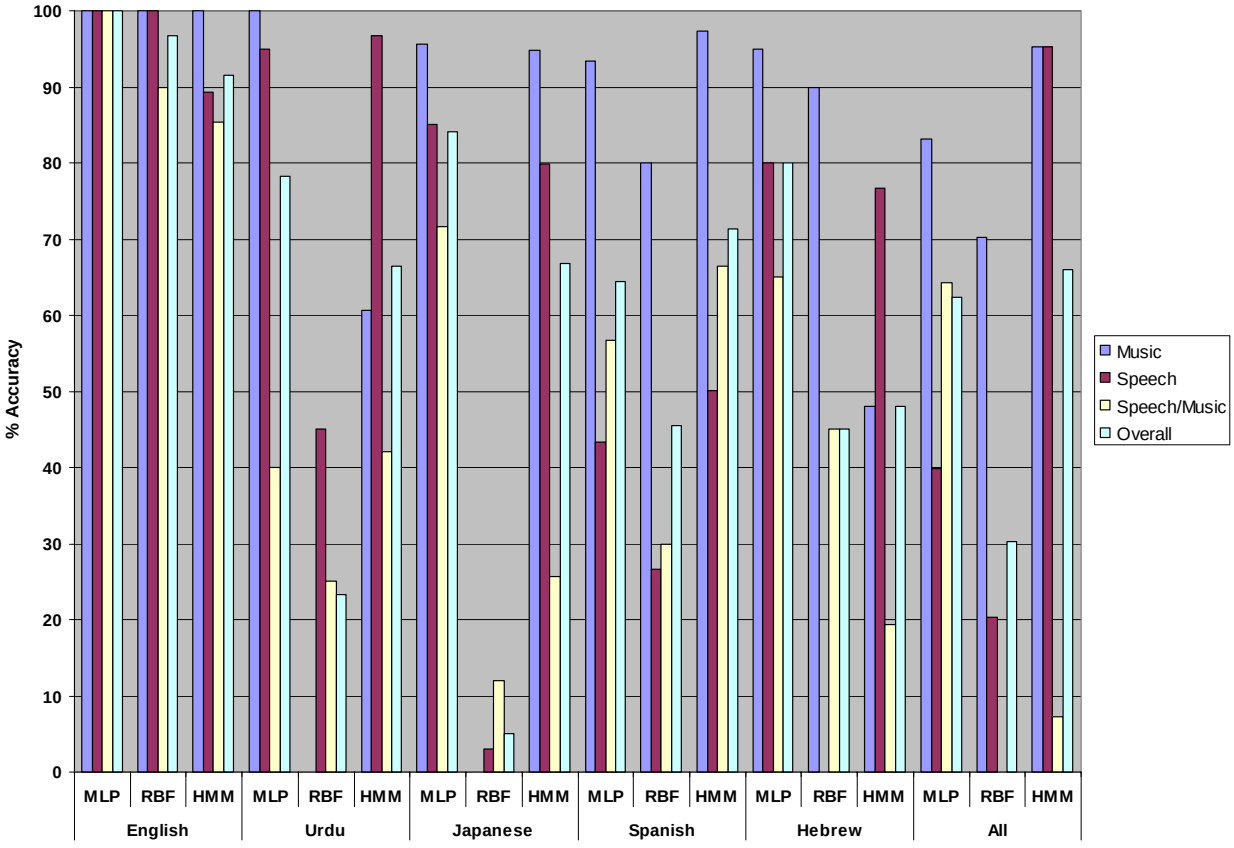

Figure 5.13: Accuracy with features: RMS-LPS, V-DWT, SF, R-ZC, LPC, and V-4MFCC

Table 5.8: Classification results for SF, R-ZC, and V-12MFCC

Language	Accuracy with MLP				Accuracy with RBF				Accuracy with HMM			
	M	S	S+M	Total	M	S	S+M	Total	M	S	S+M	Total
English	100	100	90	96.67	100	100	90	96.67	100	100	100	100
Urdu	75	95	85	85	10	55	15	26.67	33.33	80	90	67.78
Japanese	88.06	77.61	73.13	79.60	0	7.46	14.92	7.46	100	84	70	84.67
Spanish	90	53.33	70	71.11	76.67	26.67	30	44.45	91.30	69.56	50	70.29
Hebrew	90	65	85	80	80	0	55	45	26.67	50	83.33	53.33
All	72.30	50.67	69.60	64.19	70.27	22.30	0	30.86	94.60	48.65	40.10	61.12

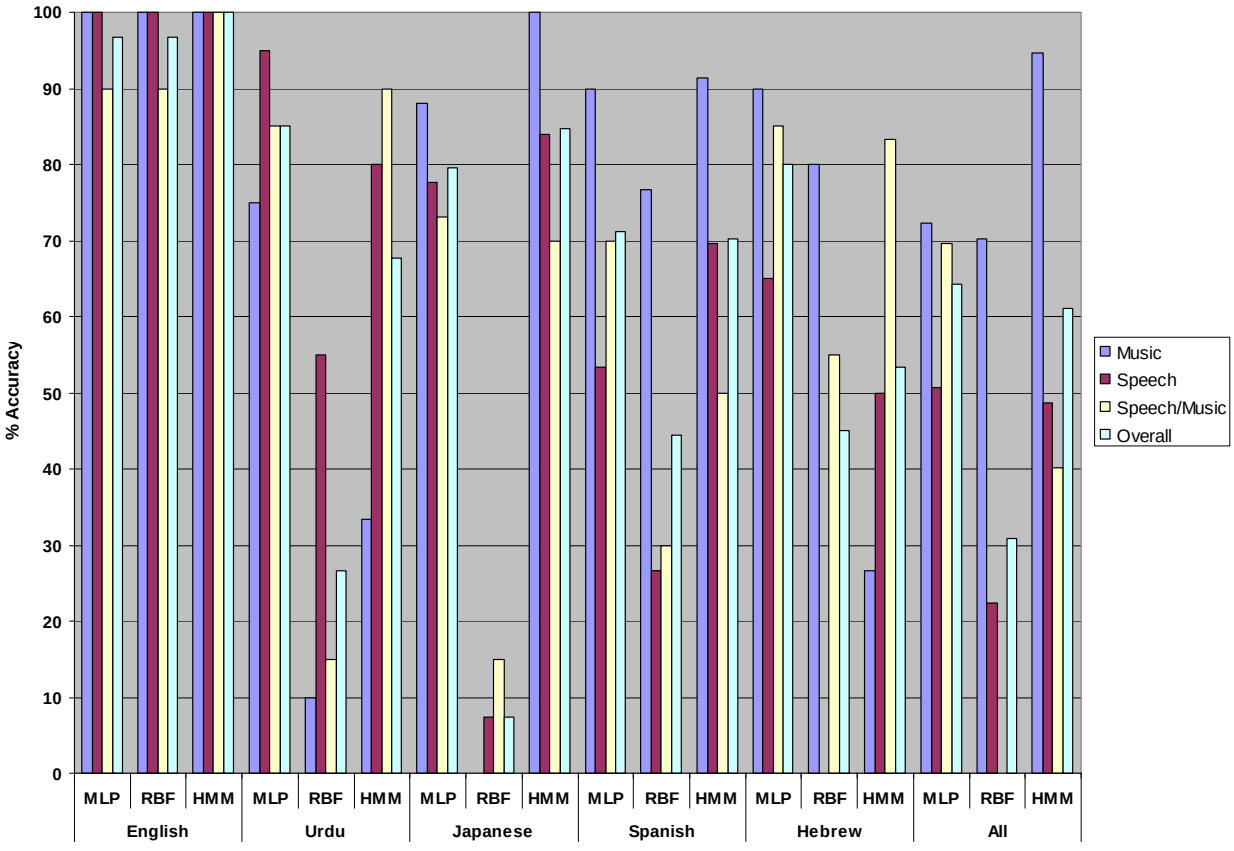

Figure 5.14: Accuracy with features: SF, R-ZC, and V-12MFCC

Table 5.9: Classification results for %LEF, R-ZC, and V-12MFCC

Language	Accuracy with MLP				Accuracy with RBF				Accuracy with HMM			
	M	S	S+M	Total	M	S	S+M	Total	M	S	S+M	Total
English	100	90	90	93.33	100	100	80	93.33	100	100	93.33	97.78
Urdu	85	95	80	86.67	46.67	60	0	35.56	63.33	86.67	86.67	78.89
Japanese	40.30	80.6	74.62	65.17	1	9	15	8.33	99	84	80	87.67
Spanish	93.33	70	43.33	68.89	80.43	41.3	34.78	52.17	76.09	71.74	84.78	77.54
Hebrew	90	60	90	80	93.33	3.33	0	32.22	30	96.67	56.67	61.11
All	79.05	64.19	47.97	63.74	69.36	38.29	0	35.88	96.40	58.12	59.46	71.33

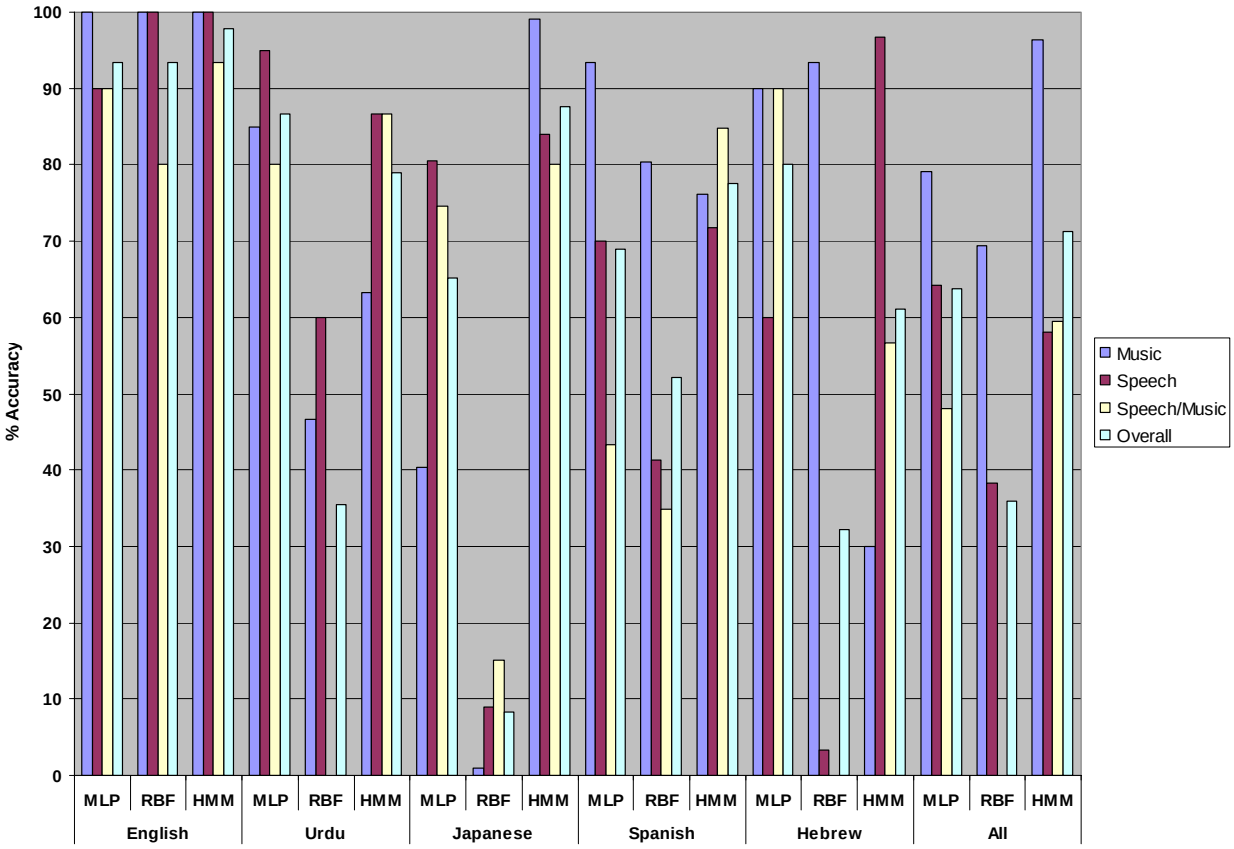

Figure 5.15: Accuracy with features: %LEF, R-ZC, and V-12MFCC

If we look at all the above tables we can say that RBF is giving satisfactory results only for English language but not for other languages. As we discussed in Section 5.1.4 on page 80 that RBF network depends on the centers of neuron and these centers must be chosen very carefully. We are using Fuzzy C-Mean clustering to find these centers. From Figures 5.16, 5.17, 5.18, 5.19, and 5.20 it is evident that the clustering results for all the languages except English are not satisfactory and since we are using the centers extracted by the same clustering technique that is why we are not getting good results for languages other than English when RBF network is used (for the comparison we have used only one feautre i.e. RMS-LPS and we expect the same behavior with other features as well).

As far as MLP and HMM are concerned, the results presented in the above tables show that both are giving good results. One of the disadvantage of HMM is that it requires long training and testing time as compared to MLP. With 35 samples for training and 15 samples for testing, HMM took around 21 sec for training and 1.3 sec for testing where as MLP took 7.3 sec for training and 0.046 sec for testing. We also have to train HMM for each audio class separately which requires more memory space. MLP is trained only once for all the audio classes simultaneously and we store only the synaptic weights which uses far less memory compared to HMM. To select a classifier for real time applications we require minimum testing time. Since MLP is giving good classification accuracy and it requires much less time and memory

than HMM that is why we have chosen MLP over HMM for investigation of long audio files.

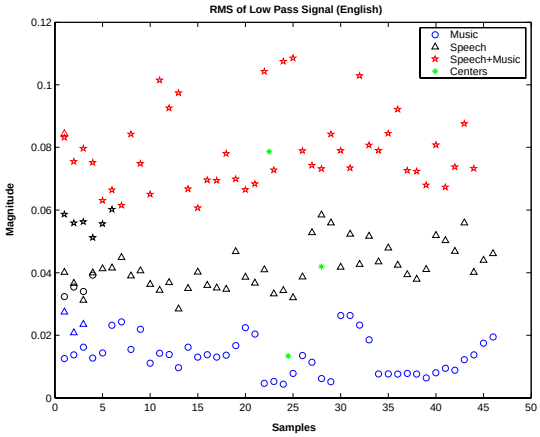

Figure 5.16: Clusters for RMS-LPS (English)

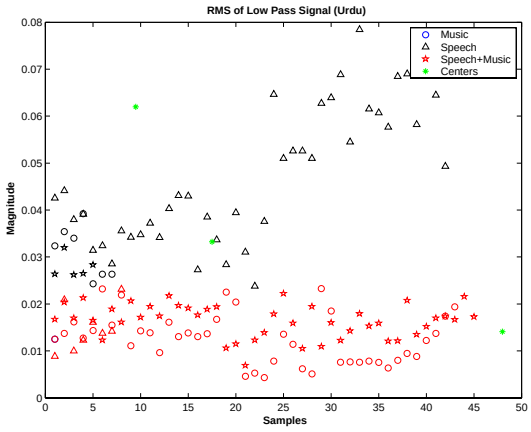

Figure 5.17: Clusters for RMS-LPS (Urdu)

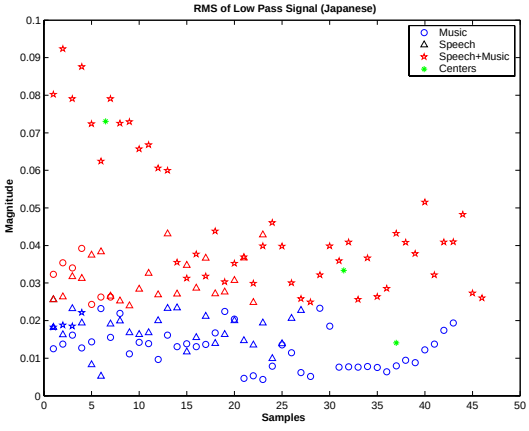

Figure 5.18: Clusters for RMS-LPS (Japanese)

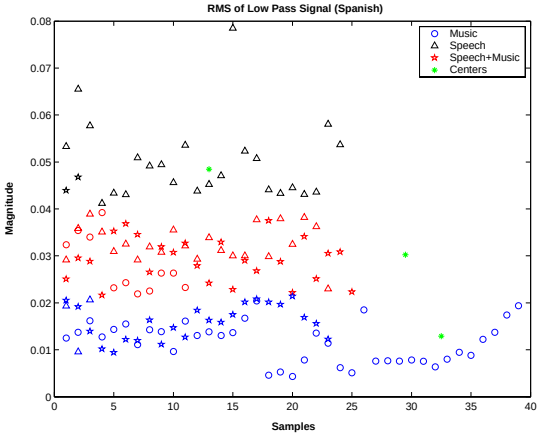

Figure 5.19: Clusters for RMS-LPS (Spanish)

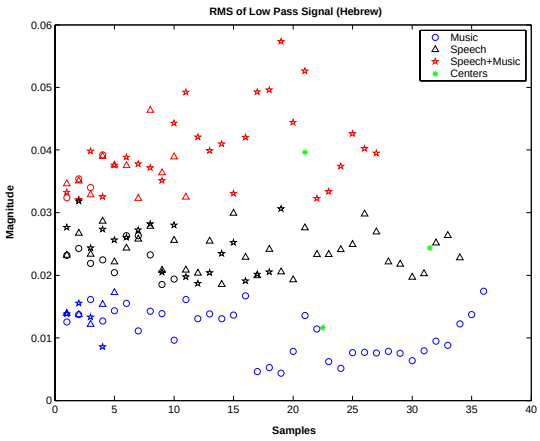

Figure 5.20: Clusters for RMS-LPS (Hebrew)

By examining the tables given above we can see that compared to English language the classification accuracy for other languages is less. It might be the case that we have different number of samples for each language, i.e. the number of samples used for training and testing. That is why the classification accuracy is different for different languages. Therefore, to investigate this factor we took the same number of samples for training and testing for all the languages and then applied MLP, RBF, and HMM as shown in Table 5.10. Since MLP is giving best results with the features: RMS-LPS, V-DWT, SF, R-ZC, LPC, and V-4MFCC that is why we have used the same features.

Table 5.10: Classification results for RMS-LPS, V-DWT, SF, R-ZC, LPC, and V-4MFCC

Language	Accuracy with MLP				Accuracy with RBF				Accuracy with HMM			
	M	S	S+M	Total	M	S	S+M	Total	M	S	S+M	Total
English	100	100	100	100	100	100	90	96.67	100.00	89.33	85.33	91.56
Urdu	100	90	80	90	100	26.67	40	55.56	84	66.67	86.67	79.11
Japanese	100	100	70	90	100	0	33.33	44.44	72	54.67	72	66.22
Spanish	100	50	70	73.33	100	40	53.33	64.44	80	92	69.33	80.44
Hebrew	100	90	40	76.67	100	6.67	33.33	46.67	82.67	40	52	58.22
All	96	58	48	67.33	24	0	0	8	85.60	33.06	79.73	66.13

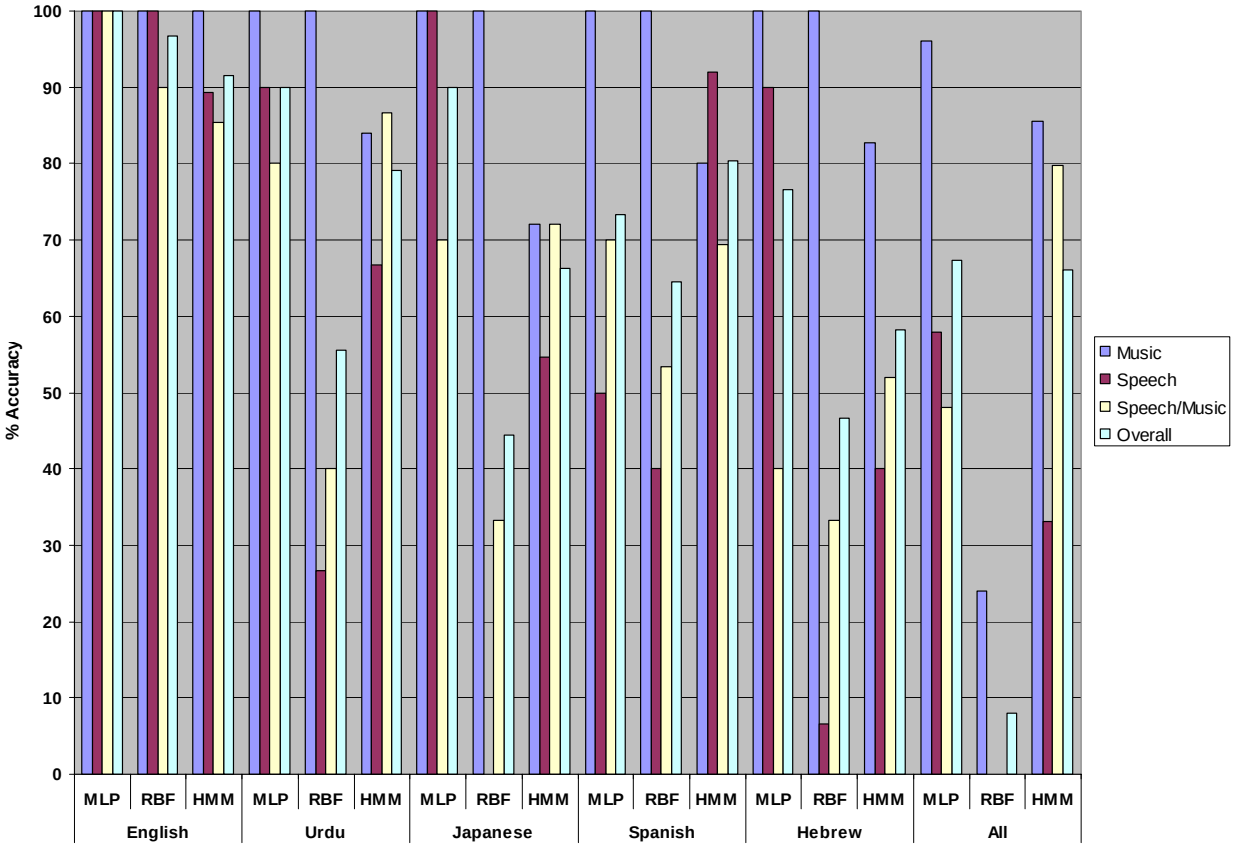

Figure 5.21: Accuracy with features: RMS-LPS, V-DWT, SF, R-ZC, LPC, and V-4MFCC

When we compare these results with the results given in Table 5.7, it is apparent that the classification accuracy for each language has increased but still it is far less than the classification accuracy of English language. So, it can be inferred that these audio features are good for English language but for other languages we have to investigate or propose some other features.

5.5 Experimental Results for Long Audio Files

So far we have only investigated audio samples each belonging to a single audio class i.e. either music, speech, or speech+music. In this section we will discuss the results when long audio files, which are longer than 3 sec are used. In the previous section we discussed that the performance of MLP is better than the other classifiers that we are using, and for the audio features that we have used in our work, English language is giving good results. That is why we have applied MLP on large audio file which contain English language content.

First, we extracted RMS-LPS, V-DWT, SF, R-ZC, LPC, and V-4MFCC from 1 sec audio samples as they were giving best results for MLP. After that we trained our MLP network of the same architecture that we have used before. For testing we have selected an audio file of duration 2 min containing music, speech, and speech+music. The features are extracted from the whole 2 min file first and then

the classification is performed. The classification is based on 1 sec clip i.e. from features of 1 sec clip we will decide to which audio class that 1 sec clip belongs and so on. We have introduced another class called "NA" i.e. if a sample or clip does not belong to any of the category then we will assign that sample or clip to NA class. The results are produced in the following table.

Table 5.11: Classification accuracy with MLP for a 2 min audio file

Accuracy with MLP				
Actual Class	***Classified As***			
	Music	Speech	Speech+Music	NA
Music	100%	0%	0%	0%
Speech	0%	46.43%	21.43%	32.14%
Speech+Music	4.84%	37.10%	32.26%	25.80%

We can see that the classification accuracy for both speech and speech+music is very low. The reason is that we have a continuous audio file and we are taking 1 sec clips for feature extraction. These 1 sec clips are not manually selected that is why it is possible that a clip contains data that belongs to more than one categories. For instance, a clip may contain music in it's first half and speech in the rest or vice versa. This degradation of classification accuracy was somehow obvious from this point of view. This is why we have introduced another class "NA". To tackle this problem and to improve the accuracy we have proposed an algorithm which after classification observes the classification result and enhances that result by removing the discrepancies in the form of "NA" class.

Proposed Algorithm

Begin

Combine all the adjacent clips belonging to the same class and add their duration.

Until all the entries are checked

If NA is in the start then check the duration of it's neighbor

If duration is greater than 1 sec then merge NA with it's neighbor

Else leave NA as it is

EndIF

EndIF

If NA is in the middle then check the class of both of its neighbors

If both belong to the same class then merge NA with it's neighbors

Else-If classes of both neighbors are not same then check the duration of both neighbors

If duration of one of it's neighbor is greater then merge NA with that neighbor

Else leave NA as it is

EndIf

EndIf

EndIf

If NA is in the end then check the duration of it's neighbor

If duration is greater than 1 sec then merge NA with it's neighbor

Else leave NA as it is

EndIF

EndIF

Return

End

After applying this algorithm on the previous classification result we finally achieved the following classification accuracy.

Table 5.12: Classification accuracy for a 2 min audio file after applying the algorithm

Accuracy After Applying the Algorithm				
Actual Class	***Classified As***			
	Music	Speech	Speech+Music	NA
Music	100%	0%	0%	0%
Speech	0%	75%	21.43%	3.57%
Speech+Music	14.51%	45.16%	37.11%	3.22%

If we compare the results in Table 5.11 with the results of Table 5.12 we can observe the improvement in the classification accuracy. The accuracy for speech has increased from 46.43% to 75% and for speech+music the accuracy has increased from 32.26% to 37.11%. The execution time of this algorithm is linear i.e. it's time complexity is $O(n)$.

Chapter 6

Software for Audio Classification

In pursuit of our research on speech/music classification we managed to create an application for such a purpose. We have developed this application in Matlab® 6.5. Figure 6.1 depicts the main interface of the application. We have three different sections of the main interface based on three different stages of classification: feature extraction, training, and testing.

6.1 Feature Extraction

This section of the interface is for feature extraction from the given audio samples as shown in Figure 6.2. The user can extract feature vectors from single audio file, either short duration or long duration, or from multiple audio files of short duration. The duration of short audio files is 3 sec and each audio file belongs to a unique

category e.g. speech, music, or speech/music. The option of feature extraction from multiple audio files is used for training and testing of the classifier only.

In this section the user has to specify the audio file(s) from which the audio features are going to be extracted, a list of audio features to be extracted as shown in Figure 6.3, and the feature filename in which the extracted feature vectors will be stored.

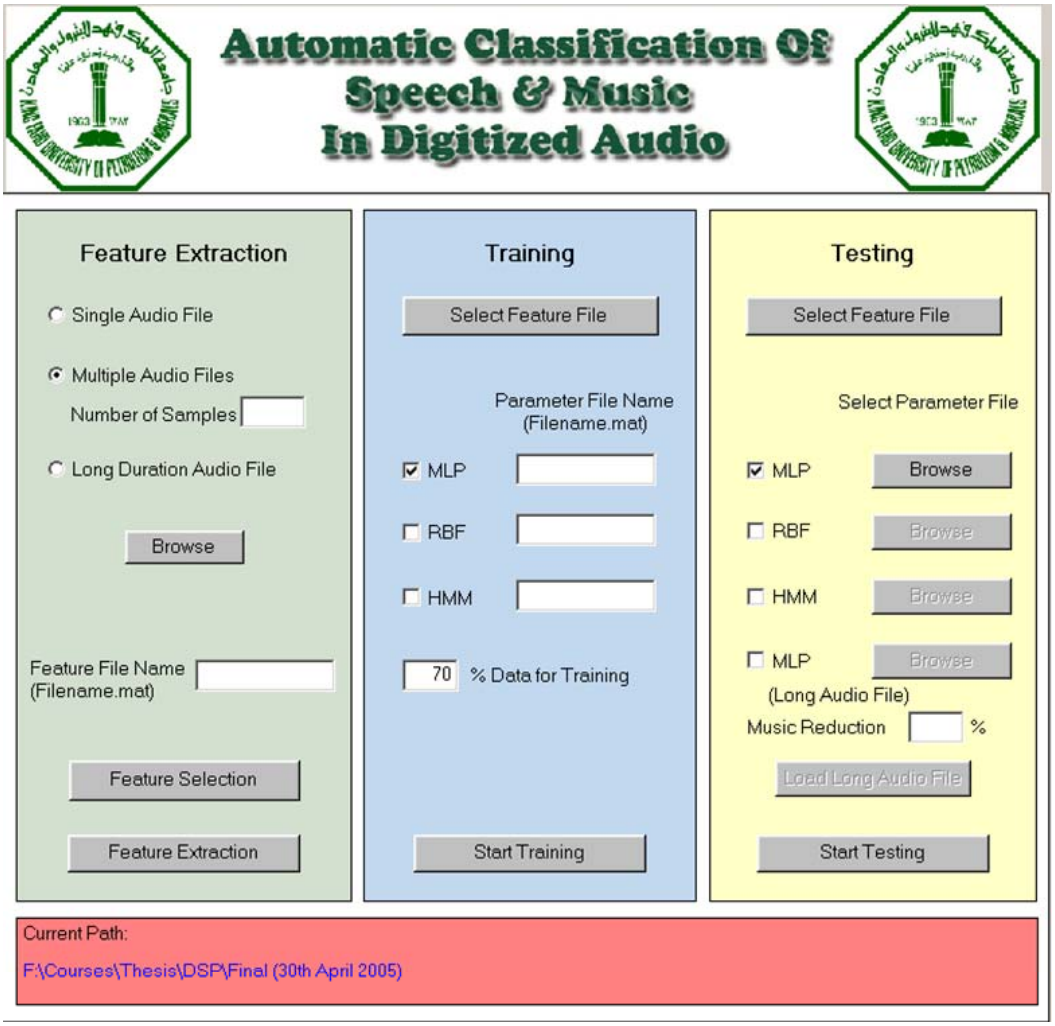

Figure 6.1: Main interface of the application

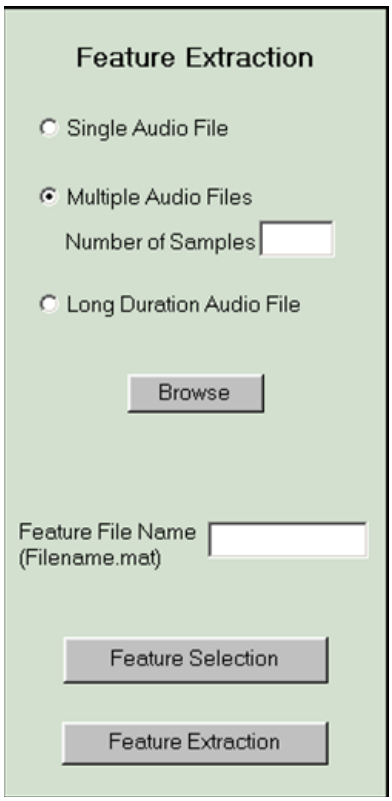

Figure 6.2: Interface for feature extraction

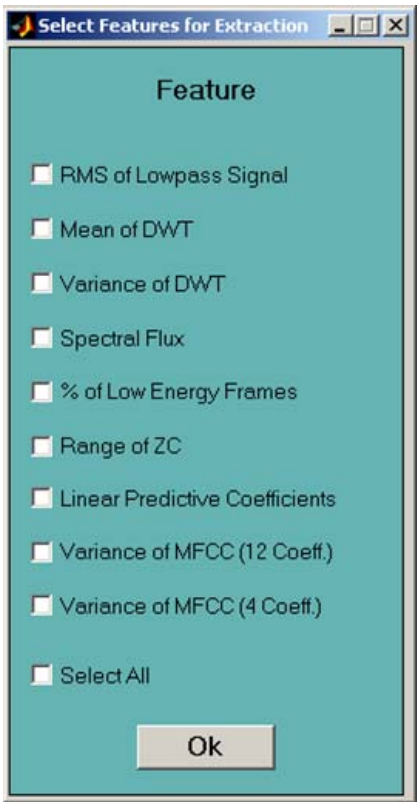

Figure 6.3: List of features

6.2 Training

This section of the main interface is used for the training of the classifier(s) as shown in Figure 6.4. After selecting the feature file saved previously, the user can select any number of classifiers for training. The user has to specify unique parameter filename for each classifier and the percentage of data to be used for training. The rest of the data will be used for the testing of the classifier(s).

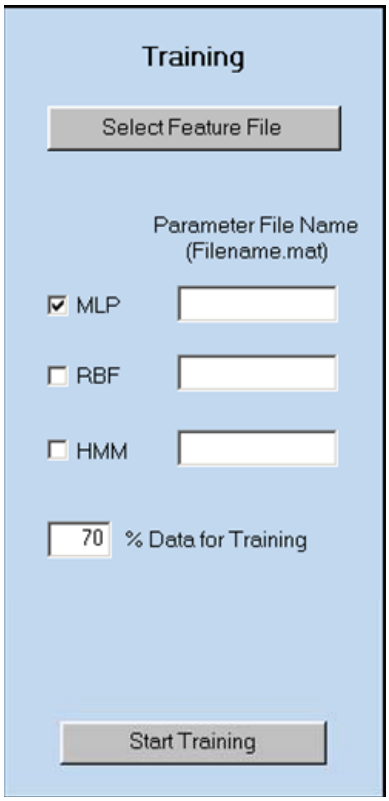

Figure 6.4: Interface for training

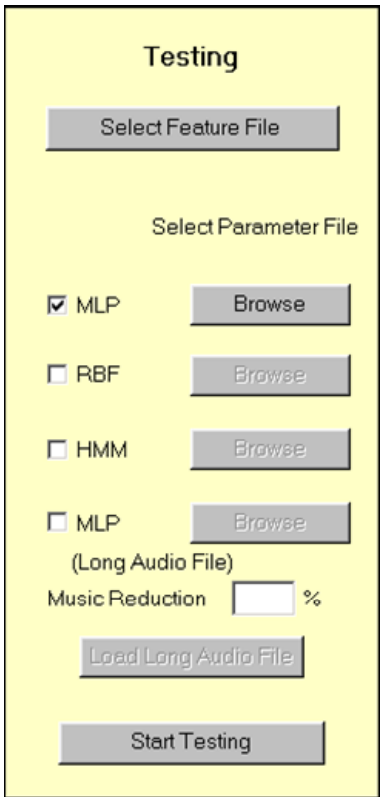

Figure 6.5: Interface for testing

Once the training is finished the training results of each classifier are presented to the user as shown in Figures 6.6, 6.7, 6.8, 6.9, 6.10, 6.11. If the user is satisfied with the results of the training of each classifier, the parameter file can be used later for testing other audio files. Otherwise the user may repeat the whole training process until the desired training results are achieved.

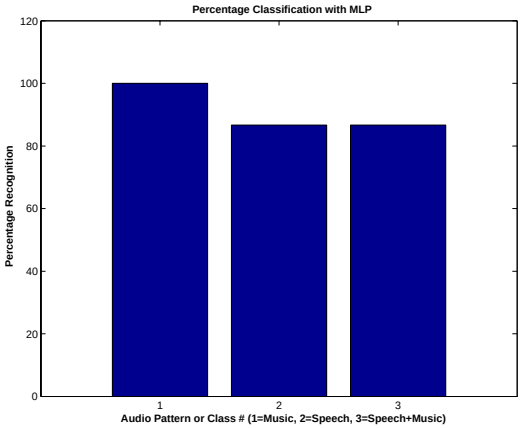

Figure 6.6: Percentage Classification Accuracy with MLP

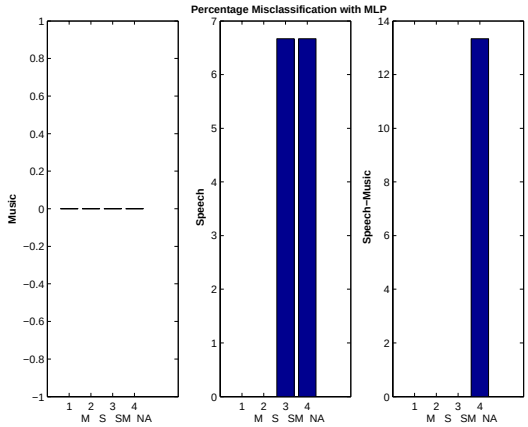

Figure 6.7: Percentage Misclassification with MLP

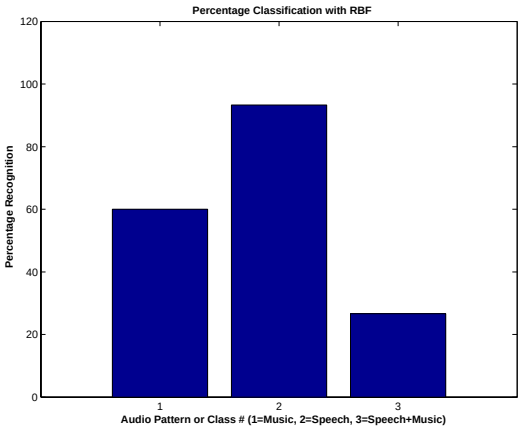

Figure 6.8: Percentage Classification Accuracy with RBF

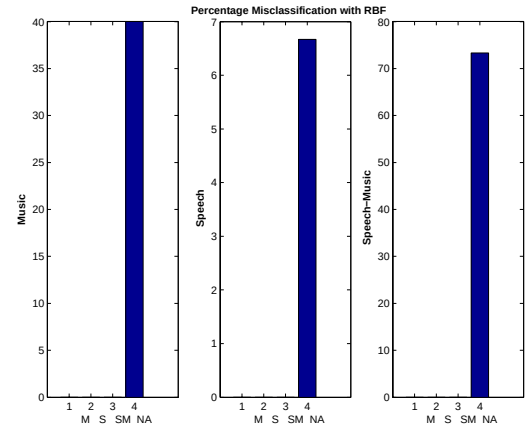

Figure 6.9: Percentage Misclassification with RBF

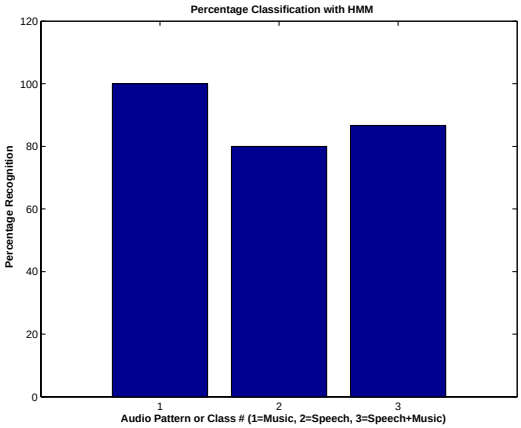

Figure 6.10: Percentage Classification Accuracy with HMM

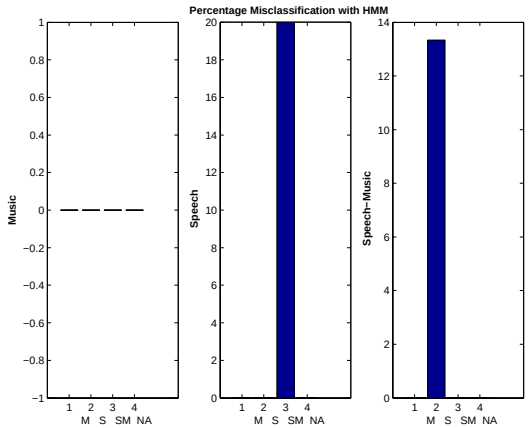

Figure 6.11: Percentage Misclassification with HMM

6.3 Testing

In this section the user can select any number of classifiers to classify the audio data of which the feature vectors were extracted previously. In this section the user can select among MLP, RBF, or HMM as a classifier if the audio data is of short duration and belongs to a single category. If the audio data is of long duration that contains data pertaining to more than one categories, then the user can only use MLP to classify that audio data. In this case the user can also specify the percentage reduction in volume of those audio segments that were classified as music. The application will create another wave file with reduced volume for music segments.

When the testing is finished the testing result of each classifier is presented to the user as shown in Figures 6.12, 6.13, 6.14. These figures presents the classification result of 150 audio files each belonging to a unique category. After finishing classification for long audio files, our proposed algorithm is then applied on the output of the classifier to improve the classification accuracy. The detail of the algorithm is already discussed in Chapter 5. If the user selects the option of classifying long duration audio file then the user will be presented with different results as shown in Figure 6.16, 6.17.

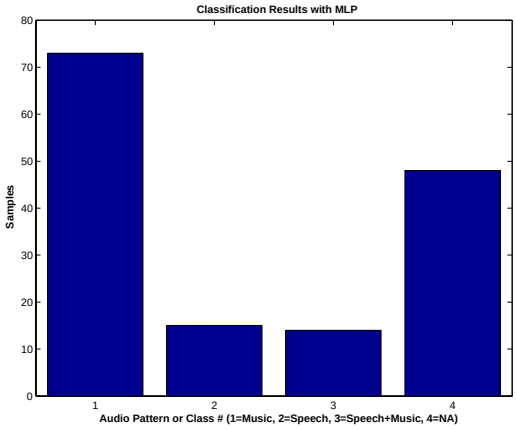

Figure 6.12: Percentage Classification Accuracy with RBF

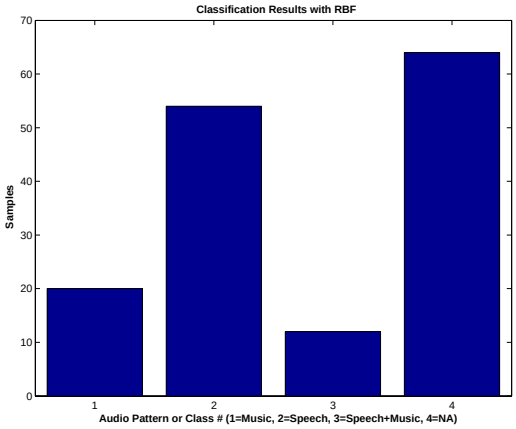

Figure 6.13: Percentage Classification Accuracy with RBF

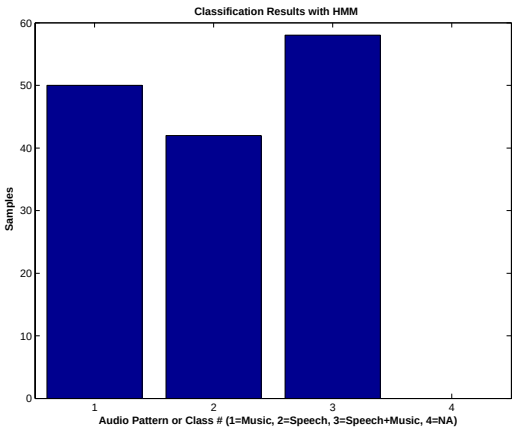

Figure 6.14: Percentage Classification Accuracy with HMM

The fourth section at the bottom of the main graphical user interface displays the current path of the application as shown in Figure 6.15. This is the location where all the Matlab files resides. The feature file will also be saved at this location. Also, the parameter file(s) saved internally by the classifier(s) while training and loaded while testing will be found here.

Current Path:

f:\courses\thesis\dsp\revised

Figure 6.15: Current path of the application

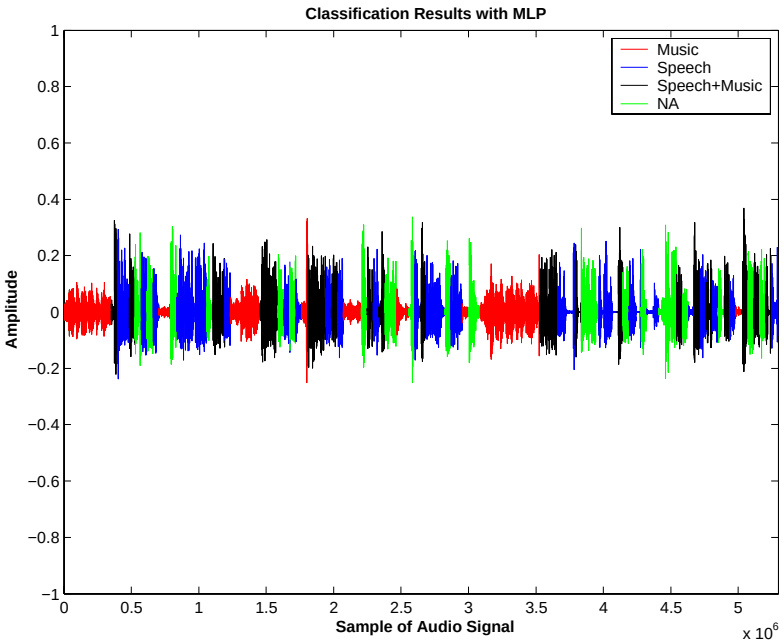

Figure 6.16: Classification Result with MLP

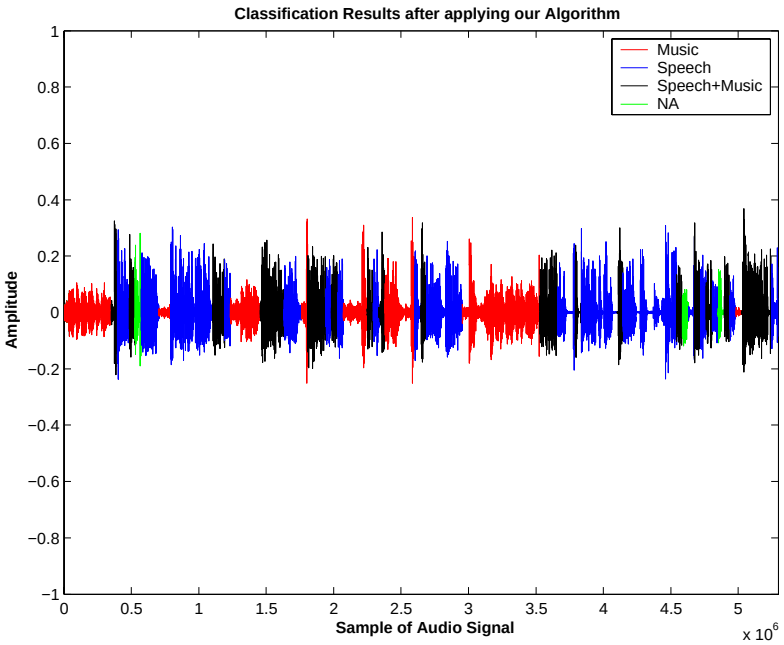

Figure 6.17: Classification Result with MLP After Applying our Algorithm

Chapter 7

Conclusion And Future Work

Many techniques have been proposed in the literature for speech/music classification. In order to achieve acceptable performance, most of them require a large amount of training data, rendering them difficult for retraining and adaptation to new conditions. Other techniques are rather context oriented, as they have been tested on specific applications, such as speech/music classification in radio programs or in the context of broadcast news transcription. In this thesis, we have conducted extensive experimentation on a diverse set of audio data using three classification frameworks and introducing five new features that have not been used earlier for music-speech classification.

The five newly proposed audio features are: Range of Zero-Crossings (R-ZC), mean of Discrete Wavelet Transform (M-DWT), variance of Discrete Wavelet Transform

(V-DWT), RMS of a Lowpass Signal (RMS-LPS), and variance of Mel Frequency Cepstral Coefficients (V-12MFCC). We have investigated these features with three other audio features: Percentage of "Low Energy" Frames (%LEF), Spectral Flux (SF), and Linear Predictive Coefficients (LPC) that have been used by many researchers. We extracted these audio features from the audio samples in five different languages containing American English, Urdu, Japanese, Spanish, and Hebrew which were extracted from documentories and movies.

The results for 3 sec duration audio samples clearly show that RBF networks give satisfactory results only for the English language. Since RBF networks depend on the centers of clusters, the results indicate that for all languages except English, the center of each cluster has not been correctly chosen by the classification algorithm.

Both MLP networks and HMMs have given good results for the same audio samples. A disadvantage of using HMMs is that it requires long training and testing time as compared to MLP. With 35 samples for training and 15 samples for testing, HMMs took close to 21 seconds for training and 1.3 seconds for testing, whereas MLPs took 7.3 seconds for training and 0.046 seconds for testing. Furthermore, HMMs need to be trained for each audio class separately, which requires more memory space. MLP is trained only once for all the audio classes simultaneously and the synaptic weights are stored once.

After investigating eight major audio features, we can conclude that applying an MLP classification framework on six of them - namely the range of zero-crossings, the variance of the Haar discrete wavelet transform, the root mean square of a low-pass signal, the spectral flux, the linear predictive coefficients, and the variance of four Mel frequency cepstral coefficients - gives the best results, achieving a 100 percent classification accuracy for English. As other languages have not achieved such accuracy, one must explore more audio features that behave similarly for different languages. Otherwise, one may need to have a closer look at different languages, closely studying their distinctive properties and the degree of similarity to music in order to justify the varying performance. There is also a need for a benchmark audio database that can be shared by researchers interested in music speech classification to facilitate more objective comparisons of various approaches.

MLP was also applied on audio samples with longer durations i.e. 2 min duration. Due to ambiguities in automatic selection of clips in the audio file the classification accuracy was affected. To improve the classification accuracy an algorithm is proposed which is applied on the classification result achieved by MLP. Our proposed algorithm has shown inspiring improvements in the results that indicate the viability of our approach.

Bibliography

[1] Michael J. Hawley. *"Structure out of Sound"*. PhD thesis, Massachusetts Institute of Technology, September 1993.

[2] John Saunders. "Real-Time Discrimination of Broadcast Speech/Music". *Proc. of the International Conference on Acoustics, Speech, and Signal Processing, Proc. ICASSP'96, IEEE*, 2:993–996, May 1996.

[3] E. M. Saad, M. I. El-Adawy, M. E. Abu-El-Wafa, and A. A. Wahba. "A Multifeature Speech/Music Discrimination System". *Proc. of the 19th National Radio Science Conference, Proc. NRSC'02, IEEE*, pages 208–213, March 2002.

[4] E. Scheirer and M. Slaney. "Construction and Evaluation of A Robust Multifeature Speech/Music Discriminator". *Proc. of the International Conference on Acoustics, Speech, and Signal Processing, Proc. ICASSP'97, IEEE*, 2:1331–1334, April 1997.

[5] M. J. Carey, E. S. Parris, and H. L. Thomas. "A Comparison of Features for Speech, Music Discrimination". *Proc. of the International Conference on Acoustics, Speech, and Signal Processing, Proc. ICASSP'99, IEEE*, 1:149–152, March 1999.

[6] E. S. Parris, M. J. Carey, and H. Lloyd-Thomas. "Feature Fusion For Music Detection". *Proc. of the European Conference on Speech Communication and Technology, EUROSPEECH'99*, pages 2191–2194, September 1999.

[7] J. Pinquier, C. Snac, and R. Andr-Obrecht. "Speech and Music Classification in Audio Documents". *Proc. of the International Conference on Acoustics, Speech, and Signal Processing, Proc. ICASSP'02, IEEE*, 4:4164–4164, May 2002.

[8] J. Pinquier, J.-L. Rouas, and R. Andr-Obrecht. "Robust Speech / Music Classification in Audio Documents". *Proc. of the 7th International Conference on Spoken Language Processing, ICSLP'02*, 3:2005–2008, September 2002.

[9] J. Pinquier, J. L. Rouas, and R. Andr-Obrecht. "A Fusion Study in Speech/Music Classification". *Proc. of the International Conference on Acoustics, Speech, and Signal Processing, Proc. ICASSP'03, IEEE*, 2:II–17–20, April 2003.

[10] Wu Chou and Liang Gu. "Robust Singing Detection In Speech/Music Discriminator Design". *Proc. of the International Conference on Acoustics, Speech, and*

Signal Processing, Proc. ICASSP'01, IEEE, 2:865–868, May 2001.

[11] H. Harb and L. Chen. "Robust Speech Music Discrimination Using Spectrum's First Order Statistics and Neural Networks". *Proc. of the 7th International Symposium on Signal Processing and Its Applications, IEEE*, 2:125–128, July 2003.

[12] H. Harb, L. Chen, and J. Y. Auloge. "Speech/Music/Silence and Gender Detection Algorithm". *Proc. of the 7th International Conference on Distributed Multimedia Systems, DMS'01*, pages 257–262, September 2001.

[13] Stefan Karnebck. "Discrimination Between Speech and Music Based on a Low Frequency Modulation Feature". *Proc. of the European Conference on Speech Communication and Technology, EUROSPEECH'01*, pages 1891–1894, September 2001.

[14] Stefan Karnebck. "Expanded Examinations of a Low Frequency Modulation Feature for Speech-Music Discrimination". *Proc. of the 7th International Conference on Spoken Language Processing, ICSLP'02*, pages 2009–2012, September 2002.

[15] W.Q. Wang, W. Gao, and D.W. Ying. "A Fast and Robust Speech/Music Discrimination Approach". *Proc. of the Information, Communications & Signal Processing, ICICS-PCM'03, IEEE*, 3:1325–1329, December 2003.

[16] K. El-Maleh, M. Klein, G. Petrucci, and P. Kabal. "Speech/Music Discrimination For Multimedia Applications". *Proc. of the International Conference on Acoustics, Speech, and Signal Processing, Proc. ICASSP'00, IEEE*, 4:2445–2448, June 2000.

[17] C. Panagiotakis and G. Tziritas. "A Speech/Music Discriminator Based On RMS And Zero-Crossings". *IEEE Transactions on Multimedia*, 2004.

[18] T. Beierholm and P. M. Baggenstoss. "Speech Music Discrimination Using Class-Specific Features". *Proc. of the 17th International Conference on Pattern Recognition, ICPR'04, IEEE*, 2:379–382, August 2004.

[19] M. M. Goodwin and J. Laroche. "A Dynamic Programming Approach to Audio Segmentation and Speech-Music Discrimination". *Proc. of the International Conference on Acoustics, Speech, and Signal Processing, Proc. ICASSP'04, IEEE*, 4:IV–309–312, May 2004.

[20] Pavel Balabko. "Speech and Music Discrimination Based on Signal Modulation Spectrum". Technical report, IDIAP Research Institute, IDIAP, Switzerland, June 1999.

[21] T. Lambrou, P. Kudumakis, R. Speller, M. Sandler, and A. Linney. "Classification of Audio Signals Using Statistical Features on Time and Wavelet Trans-

form Domains". *Proc. of the International Conference on Acoustics, Speech, and Signal Processing, Proc. ICASSP'98, IEEE*, 6:3621–3624, May 1998.

[22] C. Delfs and F. Jondral. "Classification of Transient Time-Varying Signals Using DFT and Wavelet Packet Based Methods". *Proc. of the International Conference on Acoustics, Speech, and Signal Processing, Proc. ICASSP'98, IEEE*, 3:1569–1572, May 1998.

[23] H. Ezzaidi and J. Rouat. "Speech, Music and Songs Discrimination in the Context of Handsets Variability". *Proc. of the 7th International Conference on Spoken Language Processing, ICSLP'02*, September 2002.

[24] J. D. Hoyt and H. Wechsler. "Detection of Human Speech In Structured Noise". *Proc. of the International Conference on Neural Networks, IEEE*, 7:4493–4496, July 1994.

[25] N. Mesgarani, S. Shamma, and M. Slaney. "Speech Discrimination Based On Multiscale Spectro-Temporal Modulations". *Proc. of the International Conference on Acoustics, Speech, and Signal Processing, Proc. ICASSP'04, IEEE*, 1:I–601–604, May 2004.

[26] R. Jarina, N. Murphy, N. O'Connor, and S. Marlow. "Speech-Music Discrimination From MPEG-1 Bitstream". *International Conference on Speech, Signal and Image Processing, SSIP'01, WSES Press*, pages 174–178, September 2001.

[27] R. Jarina, N. O'Connor, S. Marlow, and N. Murphy. "Rhythm Detection for Speech-Music Discrimination in MPEG Compressed Domain". *Proc. of the 14th International Conference on Digital Signal Processing, DSP'02, IEEE*, 1:129–132, July 2002.

[28] Y. Nakajima, Y. Lu, M. Sugano, A. Yoneyama, H. Yanagihara, and A. Kurematsu. "A Fast Audio Classification from MPEG Coded Data". *Proc. of the International Conference on Acoustics, Speech, and Signal Processing, Proc. ICASSP'99, IEEE*, 6:3005–3008, March 1999.

[29] D. Li, I. K. Sethi, N. Dimitrova, and T. McGee. "Classification of General Audio Data for Content-Based Retrieval". *Pattern Recognition Letters*, 22(5):533–544, April 2001.

[30] S. Esmaili, S. Krishnan, and K. Raahemifar. "Content Based Audio Classification and Retrieval Using Joint Time-Frequency Analysis". *Proc. of the International Conference on Acoustics, Speech, and Signal Processing, Proc. ICASSP'04, IEEE*, 5:V–665–668, May 2004.

[31] Xi Shao, C. Xu, and M. S. Kankanhalli. "Applying Neural Network on Content-Based Audio Classification". *Proc. of the Fourth International Conference on Information, Communications and Signal Processing, IEEE*, 3:1823–1825, December 2003.

[32] G. Tzanetakis and P. Cook. "A Framework for Audio Analysis based on Classification and Temporal Segmentation". *EUROMICRO Workshop on Music Technology and Audio Processing, IEEE*, 2:61–67, September 1999.

[33] Benjamin Kedem. "Spectral Analysis and Discrimination by Zero-Crossings". *Proceeding of the IEEE*, pages 1477–1493, 1986.

[34] L. Lu, H.-J. Zhang, and S. Z. Li. "Content-Based Audio Classification and Segmentation By Using Support Vector Machines". *ACM Multimedia Systems Journal 8*, 8(6):482–492, March 2003.

[35] S. H. Srinivasan and M. Kankanhalli. "Harmonicity and Dynamics-Based Features For Audio". *Proc. of the International Conference on Acoustics, Speech, and Signal Processing, Proc. ICASSP'04, IEEE*, 4:IV–321–324, May 2004.

[36] Vesa Peltonen. "Computational Auditory Scene Recognition". Master's thesis, Department of Information Technology, Tampere University of Technology, Finland, August 2001.

[37] G. Tzanetakis and P. Cook. "Musical Genre Classification of Audio Signals". *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, July 2002.

[38] G. Tzanetakis, G. Essl, and P. Cook. "Automatic Musical Genre Classification of Audio Signals". *Proc. of the International Symposium on Music Information Retrieval, ISMIR'01*, pages 205–210, October 2001.

[39] A. Bugatti, A. Flammini, and P. Migliorati. "Audio Classification in Speech and Music: A Comparison Between a Statistical and a Neural Approach". *EURASIP Journal on Applied Signal Processing*, 4:372–378, 2002.

[40] S. Lippens, J. P. Martens, T. De Mulder, and G. Tzanetakis. "A Comparison of Human and Automatic Musical Genre Classification". *Proc. of the International Conference on Acoustics, Speech, and Signal Processing, Proc. ICASSP'04, IEEE*, 4:IV–233–236, May 2004.

[41] Y. Wang, Z. Liu, and J. C. Huang. "Multimedia Content Analysis Using Both Audio and Visual Clues". *IEEE Signal Processing Magazine*, pages 12–36, November 2000.

[42] L. Lu, H. Jiang, and H.-J. Zhang. "A Robust Audio Classification and Segmentation Method". *Proc. of the 9th ACM International Conference on Multimedia, MM'01, ACM*, pages 203–211, October 2001.

[43] L. Lu, H.-J. Zhang, and H. Jiang. "Content Analysis for Audio Classification and Segmentation". *IEEE Transactions on Speech and Audio Processing*, 10(7):504–516, October 2002.

[44] Shahrokh Ghaemmaghami. "Audio Segmentation and Classification Based On A Selective Analysis Scheme". *Proc. of the 10th International Multimedia Modelling Conference, MMM'04, IEEE*, pages 42–48, Jan 2004.

[45] James C. Bezdek. *"Pattern Recognition with Fuzzy Objective Function Algorithms"*. Plenum Press, New York, 1981.

[46] R. O. Duda, D. G. Stork, and P. E. Hart. *"Pattern Classification"*. John Wiley & Sons, Inc., 2nd edition, 2001.

[47] Y. LeCun. "Efficient Learning and Second-order Methods". *A Tutorial at NIPS, Denver*, 1993.

[48] Simon Haykin. *"Neural Networks: A Comprehensive Foundation 2nd Ed."*. Prentice Hall, Inc., 1999.

[49] N. Gilbert and K. G. Troitzsch. *"Simulation for the Social Scientist"*. Open University Press, May 1999.

[50] G. Cybenko. "Approximation by Superpositions of a Sigmoidal Function". *Mathematics of Control, Signals, and Systems*, 2(4):303–314, 1989.

[51] L. R. Rabiner and B. H. Juang. "An Introduction to Hidden Markov Models". *IEEE ASSP Magazine*, 3(1):4–16, January 1986.

[52] Lawrence R. Rabiner. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition". *Proceedings of the IEEE*, 77(2):257–286, February 1989.

[53] Richard J. Mammone, editor. *"Artificial Neural Networks for Speech and Vision"*. Chapman & Hall Neural Computing. Chapman & Hall, 1st edition, 1994.

Vitae

- Muhammad Kashif Saeed Khan.
- Born in Hyderabad, Pakistan on September 26^{th}, 1975.
- Received Bachelor of Engineering (B.E) degree in Mechanical Engineering from N.E.D University of Engineering and Technology, Karachi, Pakistan in 2001.
- Joined King Fahd University of Petroleum and Minerals in September 2002.
- Publication: M. Kashif S. Khan, Wasfi G. Al-Khatib, M. Moinuddin, 'Automatic Classification of Speech and Music Using Neural Networks', *Proc. of the 2nd ACM International Workshop on Multimedia Databases, MMDB'04, ACM, pages 94-99, November 2004.*
- Email: kashif@ccse.kfupm.edu.sa